



**UNIVERSIDADE DO MINHO
DEPARTAMENTO DE CIÊNCIAS ECONÓMICAS E EMPRESARIAIS**

CURSO DE LICENCIATURA EM INFORMÁTICA DE GESTÃO

RELATÓRIO DE PROJETO DE LICENCIATURA

SISTEMA DE CONTROLO E RESERVA DE SALAS DA UNIVERSIDADE DO MINHO

ANO LETIVO 2012/2013 – 4º ANO

Autor: Taís Silva Andrade, N.º 1747

Mindelo, 2014

10 ANOS DE EMPENHO E DEDICAÇÃO

SISTEMA DE CONTROLO E RESERVA DE SALAS DA UNIVERSIDADE DO MINDELO

Trabalho de Conclusão de Curso para
obtenção do Grau de Licenciatura em
Informática de Gestão pela Universidade
do Mindelo.

Orientador: Samuel Lima

AGRADECIMENTOS

Agradeço a Deus pela minha vida e a todos que me rodeiam, também pela capacidade que me proporcionou em aprender.

Aos meus pais Júlia Andrade e José Andrade, que me incentivaram ao estudo, dando oportunidade para criação de todas as condições para o alcance desta jornada universitária.

Ao meu orientador Samuel Lima e ao coordenador Doutor João Dias que propiciaram a realização deste trabalho.

Aos demais professores que transmitiram seus conhecimentos, trocaram suas experiências, criando assim um elo do conhecimento.

Aos meus colegas e amigos da turma, que de forma diferente contribuíram para a realização do meu objetivo.

Ao meu noivo Tommaso Frati pelo grande apoio.

RESUMO

Este trabalho tem como objetivo apresentar um projeto para o desenvolvimento de um sistema de informação web, visando automatizar o processo de controlo de reservas de salas da Universidade do Minho. O acesso ao sistema será feito através do site da Universidade onde os utilizadores poderão cadastrar, controlar, reservar e consultar o uso das salas da Universidade.

Além disso, através do uso de padrões de projetos e de conceitos da engenharia e arquitetura de software, este trabalho tem por objetivo a construção do sistema através da tecnologia PHP – Personal Home Page e Javascript.

As tecnologias utilizadas para o desenvolvimento do sistema são gratuitas, tais como, servidor web Apache, base de dados MySQL, Microsoft Office Visio, DBDesigner, Toad for MySQL, sendo portáteis para qualquer sistema operativo, como: Linux, Windows, entre outros.

Palavras-chave: PHP, Servidor, Web, Javascript

ABSTRACT

This work has as objective to present a project for the development of a system of information web, seeking to automate the process of reserve control of the classrooms of the University. This system will be acessible through the site of the University, where users go to register in cadastre, to control, to reserve and to consult the use of classroom of the university.

Besides, through the use of design patterns and of engineering concepts and software architecture, this work also has as objective the construction of the system through the technology PHP – Personal Home Page and Javascript.

The Technologies used for the development of this system are free, such as, Web server apache, MySQL data base, Microsoft Office Visio, DBDesigner, Toad for MySQL, being carried for any operating system, as: Linux, Windows, among others.

Key words: *PHP, Server, Web, Javascript.*

ÍNDICE GERAL

AGRADECIMENTOS	IV
RESUMO	V
ABSTRACT	VI
ÍNDICE GERAL	VII
LISTA DE FIGURAS	IX
LISTA DE SIGLAS	XI
LISTA DE TABELAS	XII
CAPÍTULO 1 - INTRODUÇÃO	14
1.1 Introdução	14
1.2 Enquadramento	16
1.3 Objetivos	16
1.3.1 Objetivo geral	16
1.3.2 Objetivos Específicos	16
1.4 Metodologia	17
CAPÍTULO 2 – FUNDAMENTAÇÃO TEÓRICA	18
2.1 Sistemas de Informação	18
2.1.1 A organização e os SI	20
2.2 A Tecnologia Web	21
2.2.1 Histórico da Internet e da web	21
2.2.2 Definição da tecnologia web	22
2.2.3 A tecnologia web como plataforma para divulgação de informações	23
2.2.4 A tecnologia web como plataforma de acesso a sistemas de informação	25
2.3 Investigação Operacional	28
2.3.1 Investigação operacional: definição; história e desenvolvimento; domínios de aplicação	28
2.3.2 Tipos de modelos de investigação operacional	28
2.3.3 Fase de um estudo de investigação operacional	29
2.4 Ferramentas e Tecnologias Utilizadas	30
2.4.1 PHP	30
2.4.1.1 Método POST	31
2.4.1.2 Sessões	32
2.4.2 HTML	33
2.4.3 Base de Dados MySQL	33
2.4.3.1 Toad for MySQL 7.0	35
2.4.3.2 DBDesigner4	36
2.4.4 UML	37
2.4.5 Servidor Web apache	39
2.4.6 Javascript	41
2.4.6.1 JQuery	42
2.4.6.2 Json	44
2.4.6.3 Ajax	45

2.5 Engenharia de Software	47
2.5.1 Modelo de ciclo de vida iterativo e incremental.....	47
2.6 Segurança em sistemas de informação	49
2.6.1 Conceitos de segurança.....	49
2.6.2 Mecanismos de segurança	51
2.6.3 Ameaças à segurança	54
2.6.4 Nível de segurança.....	55
2.6.5 Políticas de segurança	56
CAPITULO 3 – CARACTERIZAÇÃO DA UNIVERSIDADE DO MINDELO	59
3.1 Dados Gerais.....	59
3.2 Visão	59
3.3 Missão	60
3.4 Valores	60
3.5 Estrutura Organizacional	61
CAPITULO 4 – DESENVOLVIMENTO DO PROTÓTIPO	62
4.1 Requisitos principais do sistema.....	62
4.1.1 Requisitos funcionais	63
4.1.2 Requisitos não funcionais:	64
4.2 Lista dos Casos de Uso	64
4.3 Análise do Sistema	65
4.3.1 Diagrama de casos de uso	65
4.3.2 Diagramas de sequência	70
4.4 Projeto do Sistema	73
4.4.1 Projeto da base de dados	73
4.4.1.1 Modelo entidade relacionamento (MER)	74
4.4.1.2 Diagrama de classes.....	79
4.5 Implementação do Sistema	79
4.5.1 Técnicas e ferramentas utilizadas	79
4.5.2 Operacionalidade da implementação	80
4.5.2.1 Interface de acesso ao sistema	80
4.5.2.2 Interface do Menu Principal	81
4.5.2.3 Cadastro de sala	81
4.5.2.4 Efetuar reserva	82
4.5.2.5 Aceitar um pedido de reserva de um horário de uma sala	83
4.5.2.6 Interface gestão de utilizadores	83
4.5.2.7 Relatório de gestão das reservas	84
4.5.2.8 Interface principal utilizador gestor de salas	84
CAPITULO 5 - CONCLUSÃO.....	86
Sugestões	86
Apêndices	91
Apêndice A – Entrevista	92
Apêndice B – Descrição estruturada dos casos de uso	93

LISTA DE FIGURAS

Figura 1 - Componentes de um SI	18
Figura 2 - Níveis de gestão	19
Figura 3 –Funcionamento da tecnologia web para acesso a um web site “tradicional”	25
Figura 4 – Utilização da tecnologia web como plataforma de acesso a outros SI.....	26
Figura 5 – SI baseado na tecnologia web	27
Figura 6 - Fases do modelo de ciclo de vida iterativo e incremental	48
Figura 7 - Gestão de utilizadores	53
Figura 8 - Organograma da UM	61
Figura 9 – Diagrama de caso de uso sistema geral do sistema	67
Figura 10 - Diagrama de caso de uso do administrador do sistema	68
Figura 11-Diagrama caso uso gestor salas	69
Figura 12 - Diagrama caso uso docente	70

Figura 13 - Diagrama sequencia efetuar reserva	70
Figura 14 - Diagrama sequencia cadastro sala	71
Figura 15-Diagrama sequencia cadastro docente	72
Figura 16 - Diagrama sequencia gerar relatorio	73
Figura 17 - Diagrama Entidade Relacionamento	75
Figura 18 - Diagrama de classes	79
Figura 19 - Interface Acesso ao sistema	80
Figura 20 - Interface menu principal	81
Figura 21 - Interface cadastro sala	82
Figura 22- Interface efetuar reserva	82
Figura 23-Interface aceitar reserva	83
Figura 24- Interface gestao utilizadores	83
Figura 25-Interface relatorio reservas	84
Figura 26-Interface principal gestor salas.....	85
Figura 27-Interface alteracao de senha	85

LISTA DE SIGLAS

AJAX	Asynchronous Javascript and XML
BD	Base de Dados
CASE	Computer Aided Software Engineering
HTML	Hypertext Markup Language
IO	Investigação Operacional
JSON	JavaScript Object Notation
MVC	Model View Control
OMG	Object Management Group
OO	Orientado por Objetos
PHP	Personal Home Page
SGBD	Sistema de Gestão de Base de Dados
SI	Sistema de Informação
SQL	Standard Query Language
UM	Universidade do Mindelo
UML	Unified Modeling Language

LISTA DE TABELAS

Tabela 1 - Requisitos funcionais	63
Tabela 2 - Requisitos não funcionais	64
Tabela 3 - Lista de casos de uso	65
Tabela 4 - Casos de uso do sistema	66
Tabela 5 - Dicionário dados tabela edifício	76
Tabela 6 - Dicionário dados tabela andar	76
Tabela 7 - Dicionário dados tabela sala	76
Tabela 8 - Dicionário dados tabela curso	77
Tabela 9 - Dicionário dados tabela disciplina.....	77
Tabela 10 - Dicionário dados tabela docente.....	77
Tabela 11 - Dicionário dados tabela horário.....	77
Tabela 12 - Dicionário dados tabela reserva.....	78
Tabela 13 - Dicionário dados edição curso.....	78

Tabela 14 - Dicionário dados disciplina edição curso	78
Tabela 15 - Dicionário dados turma	78
Tabela 16 - Descrição caso uso autenticar utilizador	93
Tabela 17 - Descrição caso uso cadastro curso	94
Tabela 18 - Descrição caso uso cadastro docente.....	95
Tabela 19 - Descrição caso uso cadastro horário.....	95
Tabela 20 - Descrição caso uso cadastro disciplina.....	96
Tabela 21 - Descrição caso uso cadastro sala.....	96
Tabela 22 - Descrição caso uso emitir relatório	97
Tabela 23 - Descrição caso uso efetuar reserva	98

CAPITULO 1 - INTRODUÇÃO

1.1 Introdução

Com o constante aumento da procura por serviços mais eficientes e uma maior preocupação na qualidade do atendimento, cada vez mais o uso de tecnologias tem sido aplicado em diversas áreas exigindo maiores esforços do profissional que convive com tal realidade. Este trabalho pretende contribuir com o avanço tecnológico da Universidade do Mindelo, apresentando um Sistema de informação web para melhorar o processo de requisições e reservas de salas para os utilizadores envolvidos.

Sistemas de Informação (SI) são um conjunto de elementos interdependentes (subsistemas), logicamente associados, para que, de sua interação, sejam geradas informações necessárias à tomada de decisões.

Segundo Rascão (2004), Sistema de Informação é um conjunto organizado de procedimentos, que, quando executados, produzem informação para apoio à tomada de decisão e ao controlo das organizações.

Quando qualquer empresa ou instituição possui um grande número de informações a serem controladas, existe quase que uma obrigação em existir sistemas informatizados de informações, os quais tornam as tarefas mais simples e as tomadas de decisões mais precisas e fundamentadas.

Os SI têm grande influência na otimização dos processos que envolvem a Universidade. Esses processos são executados por diversos sectores dentre os quais destacam-se:

- Sector financeiro – onde são controlados os dados referentes às propinas, taxas e tudo o que diz respeito à parte financeira da Universidade;

10 ANOS DE EMPENHO E DEDICAÇÃO

- Sector de recursos humanos – onde são controlados dados sobre funcionários, incluindo os professores;
- Sector Académico – onde são controlados os dados académicos, cursos e turmas.

Para organizar esse montante de informações e otimizar os processos que as envolvem, é importante a existência de sistemas informatizados.

Uma das atividades que ainda não é controlada por sistemas informáticos é o controlo de reserva de salas, sendo que, atualmente, o departamento dos serviços académicos da UM realiza, manualmente, todo o controlo de salas, tanto para as reservas semestrais das turmas quanto para reservas temporárias (empréstimos). Para as reservas semestrais, são feitos ficheiros em Excel onde são registadas as salas e os horários ocupados. Esses ficheiros são, basicamente, tabelas onde são descritas as salas e suas características, como: bloco, numero, capacidade, etc e os horários em cada dia da semana, onde são registados para quem está reservada cada sala. E para as reservas temporárias é necessário o professor deslocar-se até ao responsável pela alocação de salas para verificar a disponibilidade e assim efetuar uma reserva, ou então entrar em contacto via e-mail.

Sendo assim, este trabalho propõe-se a desenvolver um protótipo de um sistema para controlo das salas da UM, composta por partes, onde são feitos os cadastros necessários, assim como a alocação das turmas nas salas, fazer as consultas e reservas temporárias, automatizando o processo, economizando tempo e diminuindo a probabilidade de erros.

Para o desenvolvimento do sistema será utilizada a linguagem PHP e para a modelação do sistema será utilizada a linguagem UML, com suporte da ferramenta Microsoft Office Visio. E a base de dados utilizada será o MySQL, com suporte das ferramentas DBDesigner 4 e Toad for MySQL 7.0.

1.2 Enquadramento

Atualmente, não é comum encontrar um *software* específico para o controlo de requisições e reservas de salas académicas. Diante de todas as dificuldades apresentadas, identifica-se a necessidade do desenvolvimento de um aplicativo que satisfaça as necessidades dos envolvidos no processo, facilitando assim a execução e administração dos serviços realizados.

Desenvolver e implementar um sistema que solucione esse problema, principalmente dentro da realidade da Universidade do Minho, se justifica tanto no aspeto teórico por apresentar um estudo para o desenvolvimento de um sistema de informação de gestão com implementação web, quanto no aspeto pratico pela demonstração de técnicas ferramentas de *software* existentes.

1.3 Objetivos

1.3.1 Objetivo geral

O objetivo deste projeto que foi definido através de um levantamento inicial de necessidades junto a Coordenação do curso de Informática de Gestão, é projetar e desenvolver um Sistema de Informação Web que possibilite o cadastro de salas, para realizar alocação das mesmas e após controlar estas reservas, emitir relatórios sobre os dados armazenados conforme parâmetros informados. O sistema também permitira o controlo de alocação de recursos didáticos.

1.3.2 Objetivos Específicos

Como objetivos específicos temos os seguintes:

- Realizar um estudo sobre sistemas de informação e programação baseada em web;
- Realizar o levantamento de requisitos necessários para a aplicação;

- Estudar e definir as melhores tecnologias e padrões para o desenvolvimento da aplicação;
- Elaborar a documentação necessária para o desenvolvimento;
- Desenvolver o protótipo;
- Testar e implementar a aplicação.

1.4 Metodologia

Para o desenvolvimento do sistema foi necessário estudar os conceitos relacionados a metodologia. Este trabalho foi realizado em 5 etapas, a saber:

O primeiro passo foi a definição dos objetivos, bem como o estabelecimento de um plano para a execução das tarefas.

De seguida, foi feita uma recolha de informações, de modo a conhecer o funcionamento da Universidade do Minho.

Numa terceira fase, foram realizadas entrevistas com o coordenador do curso de Informática de Gestão e com o responsável pela alocação das salas.

Na quarta fase foi feita a estipulação dos requisitos, bem como a modelação dos dados.

Numa última fase foi desenvolvido o protótipo para o sistema.

A metodologia de desenvolvimento é uma atividade que se preocupa com o que o sistema tem de fazer, sendo que a técnica empregada neste projeto foi a *Unified Modeling Language*¹(UML), que é uma técnica aplicada a sistemas orientados a objetos. Nos últimos anos, a UML consagrou-se como a linguagem-padrão de modelagem adotada pela engenharia de *software*.

Os aspetos desta modelagem podem ser definidos mais adiante no capítulo II.

Também utilizou-se o modelo de desenvolvimento em cascata iterativo e incremental.

¹ Em português UML – Linguagem de modelação unificada

CAPÍTULO 2 – FUNDAMENTAÇÃO TEÓRICA

Este capítulo está organizado em três secções: A primeira argumenta sobre sistemas de informação. A segunda trata-se das ferramentas utilizadas. Por fim, a última secção trata-se da engenharia de *software*.

2.1 Sistemas de Informação

Segundo Rascao² (2004), Sistemas de Informação é um conjunto organizado de procedimentos que, quando executados, produzem informação para apoio à tomada de decisão e ao controlo das organizações.

Um Sistema de Informação (SI) é um tipo especializado de sistema e pode ser definido de diversas formas distintas. Segundo Stair e Reynolds (2006), um SI é um conjunto de componentes inter-relacionados que recolham, manipulam e disseminam dados e informações para proporcionar um mecanismo de realimentação para atingir um objetivo. A figura 1 mostra os componentes de um sistema de informação.

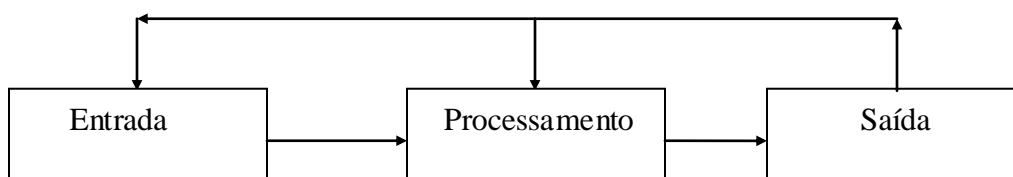


Figura 1 - Componentes de um SI

Fonte: Stair e Reynolds (2006)

² José Rascao, Sistemas de Informação para as organizações, edições silabo, 2º edição, 2004

10 ANOS DE EMPENHO E DEDICAÇÃO

Segundo Stair e Reynolds³ (2006), Um Sistema de Informação é um conjunto de componentes inter-relacionados que recolham, manipulam e disseminam dados e informações para proporcionar um mecanismo de realimentação para atingir um objetivo.

Define-se um sistema de informação como sendo um sistema que recolha, processa, armazena, analisa e dissemina informações com um determinado objetivo. Os dados são ocorrência na forma bruta, sendo necessário o agrupamento com outros dados, para obter algum valor útil. Informação é a organização de vários dados, de tal modo que, quando combinados se tornam complementares.

A arquitetura lógica de informação pode ser representada graficamente aos vários níveis da gestão, representando cada nível a visão das necessidades de informação e/ou o portfólio do negócio (unidades de negócio).



Figura 2 - Níveis de gestão

Fonte: Adotado de Rascão

³ Ralph Stair e George Reynolds, *Princípios de Sistemas de Informação*, editora Thomson, 6ª edição, 2006

2.1.1 A organização e os SI

Os SI e a tecnologia de informação têm grande importância nas organizações atuais. Eles podem alterar os processos empresariais de várias formas. Algumas delas são:

- Aumentando a capacidade das pessoas, através do fornecimento de informações, ferramentas e treinamento;
- Captando informação dos processos com o objetivo de compreensão;
- Apoiando o trabalho de gestão e melhorando a análise da informação e tomada de decisão;
- Eliminando desperdícios: eliminando papéis desnecessários, reutilizando o trabalho (por exemplo, modelos de cartas), eliminando etapas de trabalhos desnecessárias e atrasos, eliminando variações desnecessárias em procedimentos e sistemas e/ou eliminando atividades contra produtivas;
- Estruturando o trabalho de forma a promover as melhores práticas: melhorando a manipulação de dados e o trabalho geral de escritório, apoiando fluxo de trabalho e permitindo que o trabalho ocorra ininterruptamente;
- Substituindo ou reduzindo a mão-de-obra humana em um processo, seja automatizando o trabalho de projeto e/ou automatizando a manufatura;
- Integrando através de funções e de organizações: ligando fornecedores e clientes através da troca eletrônica de dados, apoiando o processo de planeamento organizacional, colaborando no projeto de produtos e através de manufatura integrada por computador e melhorando a coordenação entre tarefas e processos;
- Modificando a sequência de processo ou possibilitando o paralelismo;
- Permitindo a monitoração rigorosa da situação e objetos do processo;
- Permitindo a coordenação de processos à distância;
- Permitindo a eliminação de intermediários em um processo.

Os SI e a tecnologia da informação podem afetar a estrutura da organização, sua estratégia, suas receitas e despesas e os indivíduos que trabalham nela, além de poder promover vários

graus de mudança organizacional. Zeneti (2003) classifica os impactos das tecnologias de informação (TI) nas organizações em quatro tipos:

- Automação: a forma mais comum de mudança organizacional, em que procedimentos manuais são automatizados;
- Racionalização dos procedimentos: padronização de procedimentos operacionais, eliminando gargalos óbvios de forma que a automação possa tornar os procedimentos operacionais mais eficientes;
- Reengenharia do negócio: onde os processos são analisados, simplificados e redesenhados. A Reengenharia envolve repensar radicalmente o fluxo do trabalho e os processos de negócios usados para produzir produtos e serviços com a ideia de reduzir radicalmente os custos do negócio;
- Mudança de paradigma: radical reconcepção da natureza do negócio e a natureza da organização.

2.2 A Tecnologia Web

2.2.1 Histórico da Internet e da web

A internet surgiu como resposta à preocupação do governo americano, durante a guerra fria, de como deveria ser a comunicação militar caso ocorresse uma guerra nuclear. Numa situação como essa, as tecnologias tradicionais não funcionariam, pois um sistema centralizado poderia ser facilmente destruído. Havia, portanto, a necessidade de desenvolvimento de novas tecnologias⁴.

Em 1972, um sector do departamento de defesa americano fez a primeira demonstração pública da ARPANET, uma rede de computadores que foi precursora da Internet e, em 1983, a tecnologia da ARPANET foi substituída por uma tecnologia chamada *Transmission Control*

⁴ <http://www.teses.usp.br/teses/disponivel/12/12139/tde-14082003-104928/pt-br.php> acesso 27/11/2013 12:08

Protocol/Internet Protocol (TCP/IP), a qual era mais adequada para redes com grandes quantidades de servidores. Muitos consideram essa data como sendo o início oficial da Internet.

Ao longo das últimas décadas, várias tecnologias foram desenvolvidas na tentativa de permitir a comunicação entre computadores. Entretanto, foi a internet que atingiu este objetivo com mais sucesso, tornando-se a maior rede de computadores do mundo. Atualmente, provavelmente todas as plataformas tecnológicas permitem a utilização dos padrões da internet.

A Internet interliga várias redes e funciona de forma descentralizada, ou seja, não há controle global no nível de operações. Para pertencer à Internet, cada integrante (computador servidor) arca basicamente com os custos de suas operações, tornando-as relativamente baixos. Além disso, nenhuma mudança interna é necessária para que uma rede seja conectada à Internet.

A Internet pode ser considerada uma infraestrutura genérica de comunicação sobre a qual novas aplicações podem ser concebidas. Ao longo do tempo, vários serviços (como o e-mail, a transferência de arquivos e o acesso remoto), foram acrescentados aos padrões da Internet. No final da década de 80 e início da década de 90 um novo serviço foi criado: a *World Wide Web* (WWW).

2.2.2 Definição da tecnologia web

A *World Wide Web*, WWW, ou simplesmente *Web*, foi desenvolvida para ser “um pool de conhecimento humano, que permitisse colaboradores em locais distantes compartilhar ideias e todos os aspectos de um projeto comum”⁵. Ela deveria permitir que documentos desenvolvidos separadamente pudessem ser “ligados” facilmente e visualizados em um mesmo ambiente sem

⁵ <http://www.teses.usp.br/teses/disponivel/12/12139/tde-14082003-104928/pt-br.php> acesso 27/11/2013 12:08

que isso exigisse grandes mudanças nem que possíveis mudanças precisassem ser feitas de forma centralizada.

A tecnologia *Web* pode ser definida como um sistema de padrões que inclui:

- Padrão de endereçamento: todos os recursos da *Web* têm um endereço único e podem ser localizados de qualquer parte, independentemente da plataforma onde o recurso resida. Cada endereço é chamado de URL (*Uniform Resource Locator*);
- Padrão de comunicação: a tecnologia web utiliza um protocolo de comunicação, ou seja uma linguagem que permite a solicitação e obtenção de recursos da *Web*. Este protocolo, chamado HTTP (*HyperText Transfer Protocol*), permite a busca de recursos em diversos formatos e não somente de hipertexto como o nome sugere;
- Padrão de estruturação das informações: o padrão inicial da tecnologia *Web* para apresentação das informações estava baseado em uma linguagem de marcação chamada HTML (*Hypertext Markup Language*). Esta linguagem define principalmente elementos para a visualização de informações. Entretanto, uma extensão da tecnologia web foi a definição da metalinguagem chamada XML (*Extensible Markup Language*) a qual permite definir de forma extensível como uma informação pode ser estruturada.

A tecnologia *Web* é um conjunto de padrões para a comunicação, endereçamento e apresentação de informações, a *Web* é um conjunto formado por todas as informações e serviços (recursos computacionais) que podem ser recuperados ou utilizados através da tecnologia web.

2.2.3 A tecnologia web como plataforma para divulgação de informações

A tecnologia *Web* funciona utilizando o paradigma cliente-servidor. Neste modelo de computação, o processamento é dividido, conforme o nome sugere, entre clientes e servidores. Os clientes solicitam serviços, os quais são executados por servidores.

Na web, os clientes são *softwares* genéricos, chamados de navegadores, que proporcionam a interface com o utilizador. Os navegadores entendem os padrões da tecnologia web e são responsáveis por transformar as solicitações dos utilizadores em pedidos aos serviços web. Estes últimos recuperam os recursos (páginas) solicitados e os retornam aos clientes, que os interpretam, formatam e disponibilizam aos utilizadores.

Para recuperar uma página, os utilizadores digitam seu endereço (URL) e o navegador encaminha a solicitação ao servidor web. Portanto, para buscar uma página, só é preciso saber o seu endereço. Além disso, as páginas podem ser ligadas entre si, permitindo que o utilizador “navegue” através de várias páginas. Cada página pode conter recursos, tais como botões, figuras ou textos, os quais permitem que, quando acionados, uma nova página seja solicitada. O navegador é o responsável por converter a ativação de um recurso em uma solicitação de página.

Embora cada página web possa ter ligações para qualquer outra, comumente as páginas são agrupadas em conjunto que representam informações correlatas e ficam armazenados em um mesmo servidor *Web*. Tais conjuntos de páginas são chamados de *Web sites*.

Outra característica da tecnologia web, é que a comunicação entre o navegador e o servidor web foi concebida para funcionar sem a manutenção de conexões, ou seja, após o retorno de uma página, o servidor web não guarda informação sobre quem solicitou nem qual página foi retornada. Portanto, cada solicitação ao servidor, é independente das demais.

A figura 3 ilustra o funcionamento da tecnologia web para acesso a um web site “tradicional”.

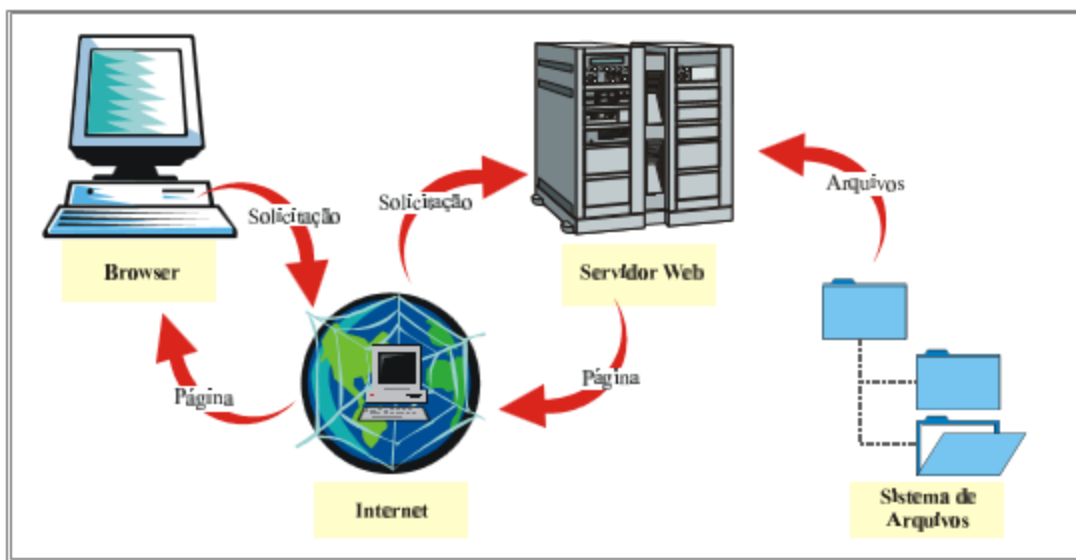


Figura 3 – Funcionamento da tecnologia web para acesso a uma web site “tradicional”

Fonte: Zeneti (2003)

No modelo de funcionamento descrito acima, as páginas web são documentos eletrônicos estáticos que permitem basicamente a divulgação de informações. Para disponibilizar novas páginas só é preciso incluir o arquivo correspondente no sistema de arquivos que ela já pode ser consultada de qualquer lugar com acesso à Internet. Este modelo de acesso simples à informação e de escala global que fez com que a tecnologia web tivesse aceitação.

2.2.4 A tecnologia web como plataforma de acesso a sistemas de informação

Ao longo do tempo, novos recursos foram acrescentados à tecnologia web. Com eles tornou-se possível:

- Enviar, junto com uma solicitação, informações ao servidor;
- Guardar “estado” entre duas chamadas ao servidor;
- Realizar processamento simples no próprio navegador;
- Desviar uma solicitação para que possa ser processada em um aplicativo no servidor, possibilitando a montagem dinâmica de páginas web; e,

- Efetuar comunicações seguras entre clientes (navegadores) e os servidores.

Os novos recursos permitem que a tecnologia web seja utilizada como infraestrutura de acesso a sistemas de informação. Dessa forma, os utilizadores interagem com os sistemas através dos próprios navegadores web, fornecendo informações aos servidores, os quais processam e geram as respostas (páginas web) dinamicamente. Assim, a troca de informações entre utilizadores e web é bidirecional de forma similar ao que ocorre com os sistemas de informação baseados nas tecnologias tradicionais.

A figura 4 ilustra o funcionamento da tecnologia web como plataforma para acesso a sistemas de informação.

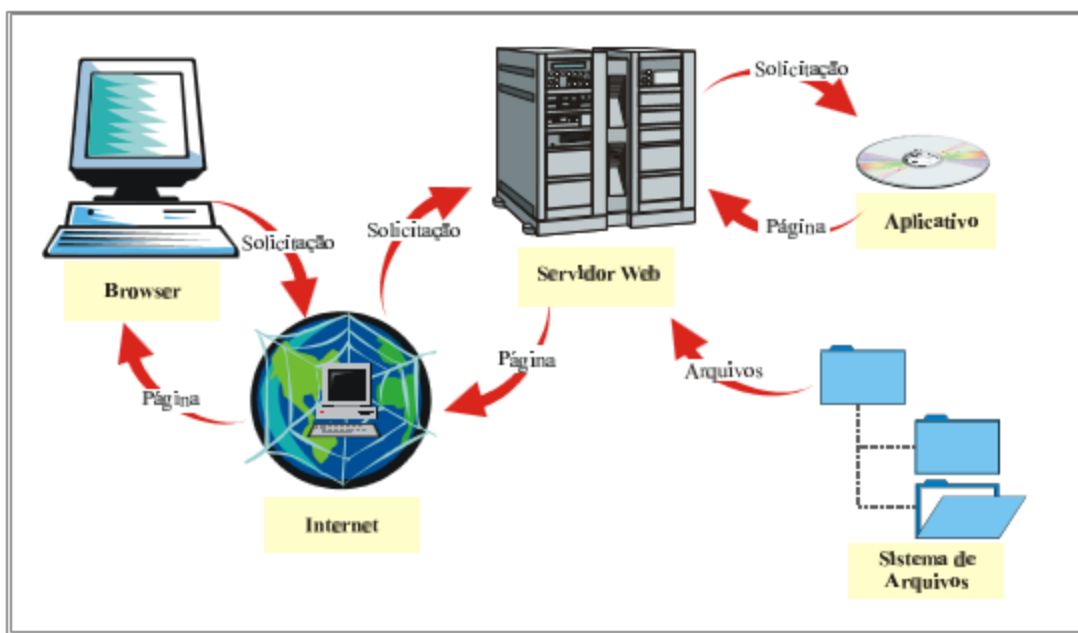


Figura 4 – Utilização da tecnologia web como plataforma de acesso a outros SI

Fonte: Zeneti (2003)

Uma limitação para a utilização da tecnologia web, conforme mostrado acima, é que ela faz intermediação entre o navegador e o aplicativo, mas caso o aplicativo precise se comunicar com outro sistema ele deve utilizar uma tecnologia tradicional. Para contornar tal restrição, novas extensões da tecnologia foram recentemente desenvolvidas, permitindo que seja usada também como infraestrutura de comunicação entre sistemas.

A figura 5 ilustra o funcionamento da tecnologia web como plataforma de comunicação entre sistemas.

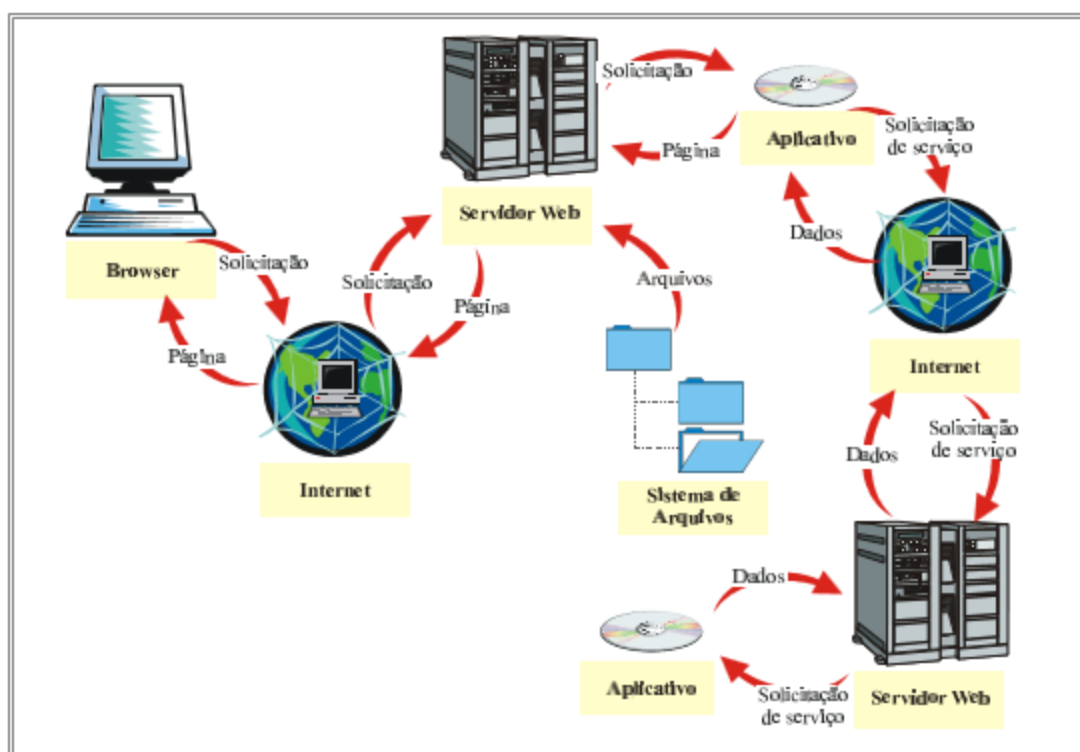


Figura 5 – SI baseado na tecnologia web 1

Fonte: Zeneti (2003)

Os sistemas de informação podem trocar informações com quaisquer outros sistemas disponíveis na web, permitindo, por exemplo, que algumas funções (ou módulos) de um sistema sejam desenvolvidas e processadas em outros lugares e por outras organizações. Em outras palavras, a tecnologia web pode ser a infraestrutura de comunicação tanto entre pessoas e sistema, como também entre os próprios sistemas. Isto elimina diversas barreiras até então existentes para a interligação entre sistemas de informação e também entre organizações.

2.3 Investigação Operacional

2.3.1 Investigação operacional: definição; história e desenvolvimento; domínios de aplicação

A Investigação Operacional procura estabelecer processos científicos de decisão que permitam uma melhor utilização de recursos escassos, ou uma melhor conceção e operação de sistemas homem-máquina, etc.

A Investigação Operacional (IO) é uma cadeira interdisciplinar. O seu início coincide com o período da II Guerra Mundial, devido à necessidade de determinar a melhor utilização dos recursos militares, que eram escassos.

As áreas de aplicação são, entre outras, no Sector de Produção (Gestão de Stocks, Planeamento de Produção, etc.), no Sector dos Serviços (Sistemas de Distribuição e Serviços), no Sector de Controlo de Qualidade (VALADARES, 2013).

2.3.2 Tipos de modelos de investigação operacional

Qualquer aplicação da IO começa por desenvolver uma representação ideal (simplificada) da realidade.

O sistema em estudo pode já ser existente, ou ser um sistema em projeto. Para um sistema já existente, pretende-se analisar o seu funcionamento, no sentido de melhorar o seu

desempenho. Para um sistema em projeto, pretende-se identificar a melhor configuração (estrutura) para o futuro sistema.

Os diferentes tipos de modelos utilizados pela IO podem agrupar-se em:

- **Simbólicos ou matemáticos**

São abstrações do mundo real. Na construção de modelos matemáticos admite-se que todas as variáveis e relevantes entre elas são quantificáveis.

- **Simulação**

Os modelos de simulação “imitam” o funcionamento do sistema. Não requerem funções matemáticas explícitas para relacionamento das variáveis, e tornam possível o estudo de sistemas complexos que não podem ser modelados ou resolvidos matematicamente.

- **Heurísticos**

Baseiam-se em regras intuitivas ou empíricas que permitem a determinação de uma solução melhorada a partir de uma dada solução inicial para o modelo.

São métodos de pesquisa que conduzem de um ponto solução para outro ponto solução, com o objetivo de melhorar o valor de critério do modelo (medida de desempenho).

2.3.3 Fase de um estudo de investigação operacional

- **Definição do problema**

- Descrição exata do objetivo do estudo
- Identificação das alternativas de decisão do sistema
- Reconhecimento das limitações e requisitos do sistema

- **Construção do modelo**

- Consiste na especificação de expressões quantitativas para o objetivo e restrições do problema em função das suas variáveis de decisão

- **Solução do modelo**

- Consiste na aplicação de algoritmos existentes ou no desenvolvimento de novos algoritmos para obtenção de uma solução ótima para o modelo

- **Validação do modelo**

- Consiste na comparação de soluções obtidas para o modelo com dados históricos do sistema, quando se trata de um sistema já existente. Para sistemas não existentes, poder-se-á utilizar um modelo de solução

- **Implementação dos resultados finais**

- O objetivo de um estudo de IO é produzir uma solução que possa e deva ser executada. Os resultados devem ser traduzidos em instruções de operações detalhadas, em forma compreensível para as pessoas que operam e administram o sistema.

2.4 Ferramentas e Tecnologias Utilizadas

Para a construção de uma aplicação web, é necessário definir uma tecnologia a ser utilizada, como servidores, linguagens de programação, bases de dados, entre outras. As tecnologias utilizadas neste trabalho encontram-se descritas ao longo deste subcapítulo.

2.4.1 PHP

Existem diversas linguagens de programação com suporte a orientação a objetos. Dentre eles, destacam-se o PHP, *Javascript*, entre outros. No nosso trabalho, foi utilizada a linguagem PHP, por ser uma linguagem de baixo custo, largamente utilizada, por possuir alta performance, excelente documentação e por ser robusta e flexível⁶

O PHP é uma linguagem de programação interpretada e, apresentando menor performance do que as linguagens de programação que são compiladas, acaba por estar orientada para projetos de web de pequenas/media dimensão, em que a velocidade de execução não seja um requisito primordial. É suportado pela maior parte dos servidores web existentes no mercado, como é o caso do apache e do Netscape, entre outros.

⁶ www.slideshare.net Acesso em 24/11/2013

O PHP é uma linguagem que permite criar *sites* web dinâmicos, possibilitando uma interação com o utilizador através de formulários, parâmetros da URL e links. A diferença de PHP com relação a linguagens semelhantes a *Javascript* é que o código PHP é executado no servidor, sendo enviado para o cliente apenas HTML puro. Desta maneira é possível interagir com a base de dados e aplicações existentes no servidor, com a vantagem de não expor o código fonte para o cliente. O código PHP fica embutido no HTML.

O PHP tem como uma das características mais importantes, o suporte a um grande número de base de dados, como dBase, interbase, mSQL, MySQL, Oracle, PostgreSQL e vários outros. Além disso, PHP tem suporte a outros serviços através de protocolos como IMAP, SNMP, NNTP, POP3 e, logicamente o protocolo HTTP.

2.4.1.1 Método POST

O método POST transfere informação via HTTP. A informação é codificada e colocado em um cabeçalho chamado QUERY_STRING.

O método POST não tem nenhuma restrição em termos de tamanho de dados a serem enviados.

O método POST pode ser usado para enviar ASCII assim como dados binários.

Os dados enviados pelo POST vão através do cabeçalho HTTP então a segurança depende do protocolo HTTP. Utilizando HTTP seguro pode-se estar seguro de que a informação está segura.

Vantagens de se utilizar o método POST:

- Variáveis enviadas na são mostradas na URL;
- Variáveis não têm limite de tamanho

Como utilizar as variáveis ⁷:

As variáveis encontram-se no vetor (array) `$_POST`, devido ao fato de terem sido passadas pelo método POST;

Independente de passar as variáveis pelo método POST ou GET, as variáveis também irão encontrar-se no vetor `$_REQUEST`, o vetor `$_REQUEST` também contém os dados do vetor `$_COOKIE`.

A nomeação das variáveis é feita dentro do formulário nos campos “*name*”, sendo a sintaxe para invocação de um item dentro de um vetor em PHP a seguinte: `NOME_DO_VETOR[“NOME_DA_VARIAVEL”]`;

2.4.1.2 Sessões

O suporte a sessões (PHP.NET,2014) no PHP consiste em uma maneira de preservar certos dados por meio de acessos subsequentes. Isto permite desenvolver aplicações mais personalizadas e que possa alternar entre paginas mantendo informações.

Um visitante ao efetuar o acesso ao seu web *site* ganha um identificador único, chamado id de sessão. Este é salvo em um *cookie* do lado do utilizador ou propagada via URL. O suporte a sessão permite registar um número arbitrário de variáveis que serão preservadas entre as requisições. Quando um visitante efetuar o acesso ao site, o PHP irá conferir automaticamente (se *session.auto_start* estiver definido como 1) ou quando pedir (explicitamente através de *session_start()* ou implicitamente através de *session_register()* quando um id de sessão específico for enviado com a requisição. Se este for o caso, o ambiente anteriormente salvo é recriado.

O sistema de controlo e reserva de salas faz o controlo em todas as páginas a existência da sessão do utilizador e sua senha e faz a sua verificação ao ser carregada, evitando assim o acesso indevido sem sua correta autenticação.

O sistema também preserva a sessão até a sessão ser finalizada.

⁷ http://www.homehost.com.br/artigos/metodo_post_envio_de_dados_invisivel_em_php-035.html

2.4.2 HTML

O HTML é uma linguagem de programação utilizada para desenvolvimento de *sites*. Esta linguagem surgiu junto com o HTTP, ambos possibilitaram a internet ser popularizada.

O HTML foi criado em 1991, por Tim Berners-Lee, no CERN (European Council for Nuclear Research) na Suíça. Inicialmente o HTML foi projetado para interligar instituições de pesquisa próximas, e compartilhar documentos com facilidade. Em 1992, foi liberada a biblioteca de desenvolvimento WWW (World wide web), uma rede de alcance mundial, que junto com o HTML proporcionou o uso mundial da web.

O HTML é uma linguagem de programação. Estas linguagens são constituídas de códigos que formam algum programa ou executam alguma ação. O HTML tem códigos para criar páginas na web. Estes códigos que definem o tipo de letra, qual o tamanho, a cor, espaçamento, e vários outros aspetos do *site*.

O HTML foi a primeira linguagem mundial, porém não é a única. Existem muitas outras linguagens destinadas à criação de páginas da web, porém o HTML ainda prevalece. Atualmente, já é possível integrar várias linguagens na mesma página da web, sendo possível usar duas ou mais linguagens no mesmo *site* (Pacievitvh, 2012).

2.4.3 Base de Dados MySQL

Segundo Damas, (2005) uma base de dados (BD) consiste numa coleção de dados estruturados, organizados e armazenados de forma persistente.

Segundo (Carriço, 1996), Um sistema de base de dados tem dois componentes fundamentais:

- A estrutura lógica e física, através da qual a informação é organizada e,

- O sistema de gestão de bases de dados (SGBD), que assegura a gestão da informação.

O *software* que gere a informação contida numa base de dados é designado por sistema de gestão de base de dados (SGBD) e constitui o *interface* entre os dados e os utilizadores. O SGBD é responsável pela implementação do sistema, ao nível do *software*, “escondendo” do utilizador as complexidades da gestão interna dos dados e possibilitando-lhe uma visão lógica da estrutura da informação, compatível com o modelo conceptual. Um Sistema de Gestão de Base de Dados (SGBD) é um grupo de programas usados como interface entre uma BD e programas de aplicação ou uma BD e o utilizador.

O principal objetivo de um SGBD é prover um ambiente adequado e eficiente para armazenar e recuperar informações de uma base de dados. A gestão de dados envolve a definição de estruturas para o armazenamento das informações e o fornecimento de ferramentas para manipulá-las.

Diversos SGBD de código livre existem, dentre eles MySQL, PostgreSQL e Berkeley DB.

Para este projeto, o SGBD utilizado foi o MySQL. Esta escolha se justifica quando se leva em conta apenas uma característica do mesmo: gestão de usuários e permissões.

O MySQL é um sistema de gestão de base de dados relacionais de código aberto usado na maioria das aplicações gratuitas para gerir bases de dados. O serviço utiliza a linguagem SQL (*Structured Query Language- Linguagem de consulta estruturada*), que é a linguagem mais popular para inserir, aceder e gerir o conteúdo armazenado numa base de dados.

Para utilizar o MySQL, é necessário instalar um servidor e uma aplicação cliente. O servidor é responsável por armazenar os dados, responder às requisições, controlar a consistência dos dados, bem como a execução de transações concomitantes, entre outras. O cliente se comunica com o servidor através do MySQL (PISO, 2012).

Características principais do MySQL

- **Gestor de base de dados**

Uma base de dados é um conjunto de dados e um gestor de base de dados é uma aplicação capaz de manipular este conjunto de dados de maneira eficiente e cómoda.

- **É uma base de dados relacional**

Uma base de dados relacional é um conjunto de dados que estão relacionados em tabelas entre as quais se estabelecem relações para manipular os dados de uma forma eficiente e segura. Para usar e gerir uma base de dados relacional se usa uma linguagem de programação SQL.

- **É Open Source**

O código fonte do MySQL se pode descarregar e está acessível a qualquer um, por outra parte usa a licença GPL para aplicações não comerciais.

- **É uma base de dados muito rápida, segura e fácil de usar**

Graças à colaboração de muitos utilizadores, a base de dados se melhorou otimizando-se em velocidade. Por isso, é uma das bases de dados mais usadas em internet.

O SQL – *Structured Query Language* é uma linguagem específica de acesso e manipulação de bases de dados relacionais.

O modelo entidade relacionamento é um dos modelos mais utilizados para representar os dados e possui três elementos básicos: entidades, atributos e relacionamentos e utiliza um diagrama entidade relacionamento (DER), que é uma notação gráfica simples, de forma a facilitar a compreensão do modelo (CHIOSSI e MORAES, 2006).

Dicionário de dados é uma listagem organizada de todos os elementos de dados que são pertinentes ao sistema, com definições precisamente rigorosas, de modo que tanto o utilizador, quanto o analista de sistemas tenham um entendimento comum das entradas, saídas, componentes de depósitos e [até] cálculos intermediários (PRESSMAN, 2002).

2.4.3.1 Toad for MySQL 7.0

Ferramenta utilizada para a implementação da base de dados. Ferramenta gráfica para manutenção das tabelas da base de dados.

Toad for MySQL (Rocha, 2009) é uma ferramenta excelente que auxilia administradores e desenvolvedores de base de dados *MySQL* a trabalharem com maior rapidez e facilidade. O programa possibilita-lhe a vantagem de trabalhar por abas. Desta maneira, mantém-se uma janela de trabalho organizada, com cada uma das suas ações devidamente separadas nestas abas.

Cria e executa *queries* (consultas) em todas as tabelas da base de dados. Cada uma destas buscas pode ser apresentada em linhas de código, com todas as sintaxes devidamente estruturadas. *Toad for MySQL* (Rocha, 2009) possibilita a verificação de cada detalhe sobre estruturas da base de dados.

2.4.3.2 DBDesigner4

Ferramenta para elaborar os relacionamentos entre as tabelas da base de dados *MySQL*.

O *DBDesigner4* (Rocha, 2009) é uma ferramenta muito importante para programadores, analistas e em especial profissionais que lidam com base de dados.

No *DBDesigner4* (Rocha, 2009), trabalha-se com modelo de estrutura, visualização de estrutura da informação armazenada em uma base de dados (por exemplo Tabelas, Índices, Relações e outras). Embora é possível armazenar dados *default* no modelo, só representa a estrutura da informação, não os próprios dados.

Criar a base de dados de acordo com o modelo pode ser exportado como um SQL ou pode ser sincronizado com uma base de dados. Ao sincronizar também é usada para alterar a base de dados automaticamente quando o modelo for alterado.

As informações do modelo são geradas em XML.

Na hora de fazer as ligações entre as tabelas, pode-se escolher o tipo, por exemplo: 1 para 1, 1 para N, entre outras. Há uma exceção para criara tabelas N para N (redundância), então na hora de criar um relacionamento assim, o programa gera uma terceira tabela que faz a

comunicação das duas, evitando assim, cometer erros no modelo e futuramente na base de dados.

A sua interface simples e descomplicada, permite fazer uma modelagem muito mais detalhada e sem muito esforço. Basta arrastar os ícones que estão na barra ao lado esquerdo e começar a desenhar as tabelas, relacionamentos e atributos.

O sistema oferecido pelo *DBDesigner* foi desenvolvido e otimizado para uso com o *MySQL*, oferecendo assim uma ferramenta competente especialmente para esses utilizadores. Todas as peculiaridades e características do *MySQL* foram incorporadas para um controlo adequado de seus conjuntos de informações.

Modelos de interface

Para trabalhar com este gerenciador, há dois modelos de interface. Um é o *Design Mode*, usado para criar e manter o modelo visual das bases de dados, o outro é o *Query Mode*, indicado para trabalhar com dados de tabelas e para criação de elementos SQL para uso em PHP, *Kylix* ou outras linguagens de programação.

Plugins e código aberto

Os modelos criados pelo *DBDesigner* são armazenados em XML, os quais podem ser modificados por *plugins* usados diretamente pelo programa. Por possuir código aberto, é fácil para programadores desenvolver novos *plugins* ou expandir este programa para suprir necessidades específicas.

Ambiente

O ambiente de trabalho do *DBDesigner* é baseado no *layout* padrão para utilitários. A navegação básica é semelhante à do Adobe *Illustrator* e do *Photoshop*, as paletas incluem o recurso “*Bird Eyes View*” (Visualização Olhos de Pássaro), tabelas, etiquetas, regiões e imagens são disponibilizadas, os recursos de soltar e arrastar (*drag-and-drop*) e de menus *popup* são constantemente disponíveis, as funções refazer e desfazer são ilimitados.

2.4.4 UML

UML é a sigla de Unified Modeling Language, que pode ser traduzido por linguagem de modelação unificada (NUNES e O'NEILL, 2004).

Segundo Videira e Silva (2005), o UML é uma linguagem para especificação, construção, visualização e documentação de artefactos de um sistema de informação. É promovido *standard* pelo Object Management Group (OMG), com contribuições de diversas empresas da indústria de software.

A ênfase da UML é na definição de uma linguagem de modelação *standard*, e por conseguinte, o UML é independente das linguagens de programação, das ferramentas CASE, bem como dos processos de desenvolvimento. O objetivo do UML, é que, dependendo do tipo de projeto, da ferramenta de suporte, ou da organização envolvida, devem ser adotados diferentes processos/metodologias, mantendo-se contudo a utilização da mesma linguagem de modelação.

A UML providencia as seguintes particularidades:

- Semântica e notação para tratar um grande número de tópicos atuais de modelação;
- Semântica para tratar tópicos de modelação futura, relacionados em particular com a tecnologia de componentes, computação distribuída, *Framework* e Internet;
- Mecanismos de extensão de modo a permitir que futuras aproximações e notações de modelação possam continuar a ser desenvolvidas sobre o UML;
- Semântica e sintaxe para facilitar a troca de modelos entre ferramentas distintas.

O UML é uma linguagem que utiliza uma notação padrão para especificar, construir, visualizar e documentar sistemas de informação orientados a objetos.

Pela abrangência e simplicidade dos conceitos utilizados, a UML facilita o desenvolvimento de um SI. Permite integrar os aspetos de natureza tecnológica, que irão constituir o sistema

informático, ajudando a dominar a complexidade das regras de negócio e definir os processos e fluxo de informação.

A UML recorre a uma notação padronizada constituída por um conjunto limitado de elementos de modelação, que podem ser tipificados em diagramas, abstrações e relacionamentos.

A UML disponibiliza o seguinte conjunto de diagramas: diagrama de casos de utilização, de classes, de sequência, de colaboração, de atividades, de objetos, de estados, de componentes e de instalação.

Abaixo serão conceituados apenas os diagramas que serão utilizados no nosso trabalho:

- **Diagrama de casos de utilização:** Descreve a relação entre atores e casos de utilização de um dado sistema. Este é o diagrama que permite dar uma visão global e de alto nível do sistema, sendo fundamental a definição correta da sua fronteira;
- **Diagrama de classes:** Descrevem a estrutura de um sistema, em particular as entidades existentes, as suas estruturas internas, e relações entre si;
- **Diagrama de sequência:** Ilustra interações entre objetos num determinado período de tempo.

2.4.5 Servidor Web apache

Um servidor Web é um *software* que permite aos utilizadores, o acesso às páginas Web, ou seja, aos arquivos no formato HTML a partir de um navegador (também chamado de browser), instalado num computador remoto.

Um servidor *Web* é, então um “simples” *software*, capaz de interpretar as consultas (ou pedidos) HTTP que chegam na porta associada ao protocolo HTTP (por padrão, porta 80), e de fornecer uma resposta com este mesmo protocolo.

Hoje em dia temos muitos exemplos de Web Server, cada um tendo um foco maior em determinada tecnologia, mas não fugindo ao seu objetivo primário que é o de prover acessibilidade a páginas Web. Entre eles, podemos destacar:

- Apache: utilizado para páginas Web que trabalhem com tecnologias Java (JSP), PHP e Ruby on Rails;
- Microsoft IIS (Internet Information System): usado para tecnologias Microsoft (ASP e ASP.NET);
- Microsoft PWS (Personal Web Server);
- Xitami.

O servidor web adotado no nosso sistema, foi o servidor *web* Apache, por ser um *software* poderoso, robusto, seguro e amplamente usado no mundo (ALECRIM, 2006).

Apache é o servidor mais conhecido da *Internet*. É uma aplicação rodando nos Sistemas Operativos do tipo Unix, mas agora, foi portado por vários Sistemas, inclusive o Microsoft Windows.

Apache foi tirado do nome de como ele foi desenvolvido (“A patchi Server” que significa um servidor remendado), pois o fruto é uma série de correções de *softwares*, para torna-lo uma solução segura. Apache é considerado seguro porque poucas de suas vulnerabilidades são publicadas.

Assim, assim que falhas de segurança são detetadas, elas são corrigidas imediatamente e uma nova versão do aplicativo é publicado.

Porque escolher o apache como servidor web?

Ao se decidir a colocar um servidor web *on-line*, há uma série de pacotes de softwares disponíveis a partir do qual a escolher. Eu acredito fortemente que a melhor escolha para um servidor web é o servidor web Apache.

Como uma fonte aberta, o Apache é um *software* livre. Obviamente, o custo é muitas vezes um fator para decidir sobre um produto para usar.

Apache também ganha em recursos. Apache suporta módulos carregáveis, o que torna capaz de executar praticamente qualquer linguagem de script dinâmica disponível, incluindo Perl, *Python*, PHP e muito mais.

Apache também é extremamente estável. Segundo a pesquisa de Netcraft⁸, o apache atende a mais de 55.26 % de todos os *sites* da internet de hoje.

Apache também é extremamente portátil. Atualmente, o servidor Apache é executado em uma grande variedade de plataformas, incluindo Mac, Windows, Solaris, Linux, AIX, e muitos mais.

Finalmente, o Apache tem alguns dos melhores suporte disponível. Sua ampla utilização resultou em uma variedade de opções de suporte. Pode-se comprar planos de suporte pagos a partir de organizações de serviço ou pode-se facilmente encontrar livre em comunidades de suporte *on-line*.

2.4.6 Javascript

JavaScript foi desenvolvido por Brendan Eich, da Netscape Communications Corporations.

JavaScript é uma linguagem de *script* que possibilita adicionar um novo nível de interatividade e função às páginas web.

⁸ <http://news.netcraft.com/archives/2013/01/07/january-2013-web-server-survey-2.html> acesso 25/11/2013 14:55

Um script é uma sequência de instruções (ou, até mesmo um programa) que são interpretados ou executados por um outro programa e não pelo processador do computador. Eles são mais fáceis e mais rápidos de escrever do que os programas e necessitam de um editor de texto simples, que pode ser gratuito⁹.

Diretamente de HTML, se houver erros nos scripts, o programa os interromperá e poderá até travar o computador. Normalmente pode-se visualizar a mensagem de erro gerado no navegador.

2.4.6.1 JQuery

No início da web, a linguagem *JavaScript* era vista como a linguagem difícil de aprender e utilizar. Depois de estudos da linguagem, foi escrito um script que validava um formulário e que com certeza daria erros nos navegadores que eram diferentes do que foi utilizado para o desenvolver. Afinal, cada navegador tinha o seu próprio padrão de implementação do *JavaScript*.

Foram estabelecidos padrões para a linguagem e também houve o surgimento de diversas bibliotecas que foram criadas para lidar com essas diferenças entre navegadores. Desse universo é que surge a JQuery¹⁰.

A JQuery é uma biblioteca JavaScript, cross-browser, desenvolvida por John Resig. Foi lançada em Janeiro de 2006, e se tornou uma das bibliotecas JavaScript mais populares da internet.

A biblioteca é disponibilizada como *software* livre e disponibilizada sobre regras das licenças MIT e GPL. Ou seja, é uma biblioteca que pode ser utilizada tanto para projetos pessoais como para projetos comerciais, sem a necessidade de pagar qualquer tipo de licença de uso.

Foi criada tendo como foco a simplicidade e com o objetivo de facilitar a programação JavaScript. Programas que utilizariam linhas e mais linhas de código para obter um determinado efeito ou carregar um componente AJAX, são substituídos por poucas instruções

⁹ <http://apostilando.net.br/downloads/3343apostilajavascript.pdf> acesso 11/02/2014 às 18:35

¹⁰ http://www-l3tol.com/emp/introducao_jquery.pdf acesso em 11/02/2014 às 19:15

utilizando JQuery o que permite o uso dela por *designers* e desenvolvedores com pouco conhecimento de JavaScript.

A JQuery é leve (em torno de 30 kb), extensível, oferece suporte a *plugins* e atualmente conta com uma vasta comunidade de programadores que diariamente acrescentam novos recursos à biblioteca. Além de funções, oferece ao programador uma grande quantidade de controlos de interface.

Para que serve?

A JQuery pode ser utilizada para:

- Adicionar efeitos visuais e animações;
- Ter acesso e manipular o DOM;
- Carregar componentes AJAX;
- Prover interatividade;
- Alterar conteúdos;
- Simplificar tarefas JavaScript.

Compatibilidade

A JQuery foi criada para ser uma biblioteca compatível com qualquer navegador web. Ela resolve para o programador a penosa tarefa de desenvolver programas em JavaScript tendo que prever a enorme gama de navegadores em que sua página poderá rodar. Sabemos que cada navegador tem seu próprio conjunto de características de implementação, que ainda podem variar de acordo com a plataforma e sistema operativo onde estiver rodando. Com a JQuery, a programação é única e transparente.

A JQuery oferece suporte a CSS3. Pode-se utilizar os seletores CSS3 mesmo que o navegador não tenha suporte a esta versão de folhas de estilo. No caso da JQuery, ela própria implementa os seletores CSS3, o que a torna independente do navegador em que estiver rodando.

Obtendo e instalando a JQuery

A query pode ser obtida gratuitamente no site <http://jquery.com/>. Na página principal, no lado direito da página há um botão de *download* (jQuery) onde podemos escolher entre duas versões:

- **PRODUCTION:** a versão voltada para ambiente de produção. É uma versão com o código em formato minimizado, sem quebras de linha e com o código obscurecido. Essa versão tem em torno de 15% do tamanho da versão de Desenvolvimento (DEVELOPMENT).
- **DEVELOPMENT:** a versão de desenvolvimento. É a versão mais compactada da biblioteca e com o código não obscurecido. É ideal para ambientes de desenvolvimento por se integrar facilmente às ferramentas IDEs.

Basta escolher a versão que se deseja instalar e clicar em Download. A biblioteca abrirá em formato de JavaScript no próprio navegador, e depois clica-se em Arquivo> Salvar e guardar a biblioteca na pasta de desenvolvimento.

Usualmente a biblioteca é guardada com o nome `jquery-1.10.4.min.js` quando for a versão compactada, ou com o nome `jquery-1.10.4.js` quando for a versão de desenvolvimento.

2.4.6.2 Json

JSON é basicamente um formato leve de troca de informações/dados entre sistemas. Mas JSON significa JavaScript Object Notation¹¹.

O JSON além de ser um formato leve para troca de dados, é também muito simples de ler.

JSON é uma ferramenta de formatação de troca de dados que possui um formato de dados compacto e nativo de JavaScript. A JSON nada mais é do que uma maneira de representar os objetos em JavaScript. Para trabalhar com esse formato não é necessário usar DOM (*Document Object Model*) ou qualquer *framework* ou *kit* específico, embora muitos dos

¹¹ <http://www.devmedia.com.br/introducao-json/23166> acesso em 05/03/2014 às 11:25

frameworks atuais forneçam algum tipo de suporte à JSON. JSON assemelha-se com as matrizes do JavaScript, mas pode armazenar os mesmos dados que os documentos XML. JSON é constituído em duas estruturas:

- Uma seleção de pares (nome e valor). Em várias linguagens isto é caracterizado como um objeto, *struct*, dicionário, *hash*, tabela, *keyed list* ou *arrays* associativos.
- Uma lista ordenada de valores. Na maioria das linguagens isto é caracterizado como uma *array*, vetor, lista ou sequência.

Estas são estruturas de dados universais. Virtualmente todas as linguagens de programação modernas as suportam, de uma forma ou de outras. A escolha pela utilização do JSON foi feita por ele ser mais leve que o XML (eXtensible Markup Language), portanto um arquivo de 300k em XML pode virar um arquivo de 20k ou 30k em JSON e nem por isso o arquivo JSON é difícil de um humano ler, outro motivo é que a sintaxe do JSON é muito simples e muito fácil de criar uma implementação do mesmo para qualquer linguagem.

2.4.6.3 Ajax

Ajax acrónimo em língua inglesa de Asynchronous Javascript And XML¹².

É um conjunto de tecnologia:

- Objeto capaz de fazer requisições assíncronas;
- HTML;
- JavaScript;
- XML eXtensible Markup Language;
- Json JavaScript Object Notation;
- DHTML;
- CSS.

¹² www.apostilando.org/baixar/apostila-ajax-ajax-tutorial/ acesso em 05/03/2014 às 11:15

O AJAX não é somente um modelo, é também uma iniciativa na construção de aplicações web mais dinâmicas.

O AJAX serve para:

- Criar Web *sites* mais dinâmicos;
- Acelerar o acesso à informação;
- Diminuir o tráfego de informações entre Cliente e Servidor;
- Produzir aplicações Web mais semelhantes as Desktop.

Métodos e atributos

- Métodos:
 - Open (mode, url, boolean), mode é POST ou GET, url é o endereço, pode ser relativo, o último parâmetro é TRUE para conexão assíncrona e FALSE para conexão síncrona;
 - Send() é o método SEND que ativa a conexão e faz a requisição de informações ao documento aberto pelo método OPEN;
- Atributos:
 - Status: *status* do retorno do html, são os códigos padrões do html 200 ok, 400 not found;
 - Response Text: Retorna a cadeia de caracteres que o servidor enviou;
 - Onreadystatechange: Define qual função será chamada para fazer a manipulação dos dados assim que houver um retorno;
 - readyState: Código que diz o *status* da solicitação:
 - 0 (uninitialized);
 - 1 (a carregar);
 - 2 (carregado);
 - 3 (interativo);
 - 4 (completo).

Vantagens

- Como a modificação das informações da tela são parciais, uma grande quantidade de informações deixa de trafegar inutilmente pela rede;
- O servidor que roda a aplicação fica menor carregado pois existe uma divisão de tarefas com o cliente;
- AJAX não é uma tecnologia, por isso não é necessário pagar para utilizá-la.

Problemas

- AJAX não é a solução milagrosa para todos os males e se usado de forma inadequada pode piorar uma situação que já era feia;
- Os navegadores usam diferentes métodos por isso temos que estar atentos;
- Os navegadores em geral não suportam uma carga muito pesada de scripts;
- Os botões voltar, avançar e histórico não funcionam muito bem com AJAX.

2.5 Engenharia de Software

Softwares são instruções (programas de computadores) que quando executados fornecem a função e o desempenho desejados, estruturas de dados que permitem aos programas manipular adequadamente a informação e documentos que descrevem a operação e o uso dos programas.

Segundo Pressman, (2002), a engenharia de *software* é a criação e a utilização de sólidos princípios de engenharia a fim de obter software de maneira económica, que seja confiável e que trabalhe eficientemente em máquinas reais.

Modelo de engenharia de *software* é uma abordagem estruturada para o desenvolvimento de software e tem como objetivo a simplificação do processo de fabricação de software de alto padrão de qualidade, tudo isso sem abrir mão de uma relação custo-benefício (PÁDUA, 2009).

2.5.1 Modelo de ciclo de vida iterativo e incremental

O modelo de ciclo de vida iterativo e incremental foi proposto justamente para ser a resposta aos problemas encontrados no modelo em cascata. Um processo de desenvolvimento, segundo essa abordagem, divide o desenvolvimento de um produto de *software* em ciclos. Em cada ciclo de desenvolvimento, podem ser identificadas as fases de análise, projeto, implementação e testes.

Essa característica contrasta com a abordagem clássica, na qual as fases de análise, projeto, implementação e testes são realizados uma única vez.

No Modelo de Ciclo de vida iterativo e incremental, um sistema de *software* é desenvolvido em vários passos similares (iterativo). Em cada passo, o sistema é estendido com mais funcionalidades (incremental) (LINHA DE CODIGO, 2013).

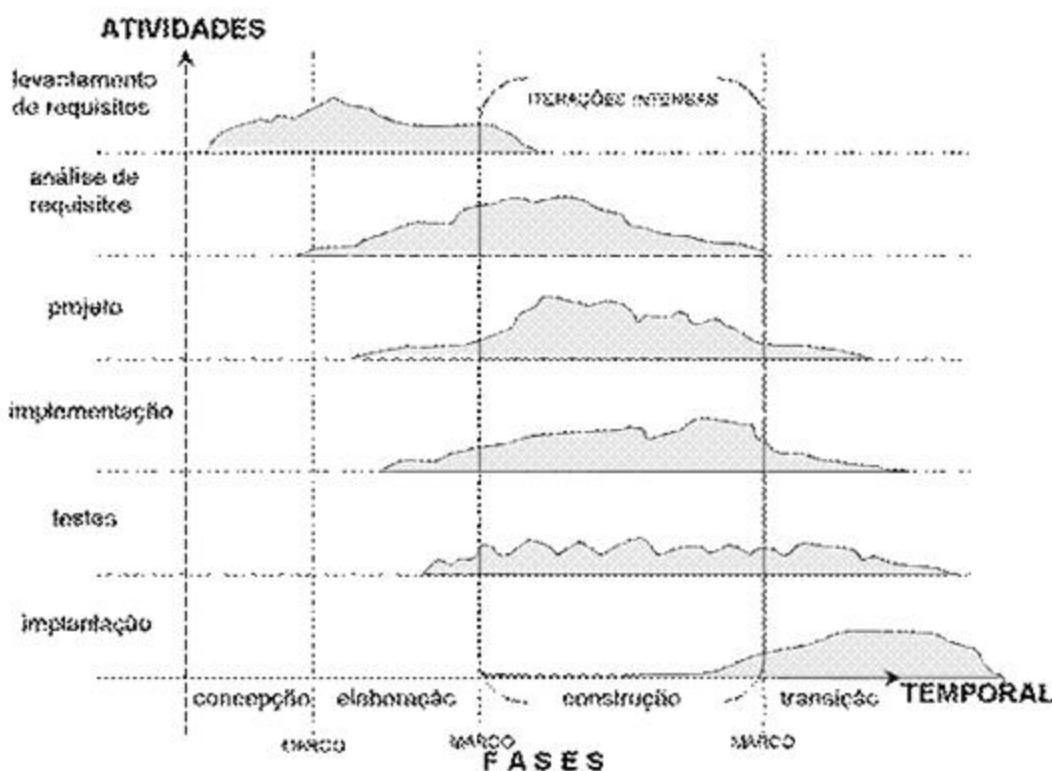


Figura 6 - Fases modelo ciclo vida iterativo e incremental

Fonte: Linha de código (2013)

2.6 Segurança em sistemas de informação

Segurança da Informação está relacionada com proteção de um conjunto de dados, no sentido de preservar o valor que possuem para um indivíduo ou uma organização.

São características básicas da segurança da informação os atributos de confidencialidade, integridade e disponibilidade, não estando esta segurança restrita somente a sistemas computacionais, informações eletrónicas ou sistemas de armazenamento. O conceito se aplica a todos os aspetos de proteção de informações e dados. O conceito de Segurança Informática ou Segurança de Computadores está intimamente relacionado com o de Segurança da Informação, incluindo não apenas a segurança dos dados/informação, mas também a dos sistemas em si.

Atualmente o conceito de Segurança da Informação está padronizado pela norma ISO/IEC 17799:2005, influenciada pelo padrão inglês (British Standard) BS 7799. A série de normas ISO/IEC 27000 foram reservadas para tratar de padrões de Segurança da Informação, incluindo a complementação ao trabalho original do padrão inglês. A ISO/IEC 27002:2005 continua sendo considerada formalmente como 17799:2005 para fins históricos.

2.6.1 Conceitos de segurança

A Segurança da Informação se refere à proteção existente sobre as informações de uma determinada empresa ou pessoa, isto é, aplica-se tanto as informações corporativas quanto às pessoais. Entende-se por informação todo e qualquer conteúdo ou dado que tenha valor para alguma organização ou pessoa. Ela pode estar guardada para uso restrito ou exposta ao público para consulta ou aquisição.

Podem ser estabelecidas métricas (com o uso ou não de ferramentas) para a definição do nível de segurança existente e, com isto, serem estabelecidas as bases para análise da melhoria ou piora da situação de segurança existente. A segurança de uma determinada informação pode ser afetada por fatores comportamentais e de uso de quem se utiliza dela, pelo ambiente ou

infraestrutura que a cerca ou por pessoas mal-intencionadas que têm o objetivo de furtar, destruir ou modificar tal informação (FERREIRA, 2013).

A CIA (Confidentiality, Integrity and Availability), Confidencialidade, Integridade e Disponibilidade representa os principais atributos que, atualmente, orientam a análise, o planeamento e a implementação da segurança para um determinado grupo de informações que se deseja proteger. Outro atributo importante é a autenticidade. Com o evoluir do comércio eletrónico e da sociedade da informação, a privacidade é também uma grande preocupação.

Os atributos básicos (segundo os padrões internacionais) são os seguintes (FERREIRA, 20113):

- Confidencialidade - propriedade que limita o acesso a informação tão-somente às entidades legítimas, ou seja, àquelas autorizadas pelo proprietário da informação;
- Integridade - propriedade que garante que a informação manipulada mantenha todas as características originais estabelecidas pelo proprietário da informação, incluindo controlo de mudanças e garantia do seu ciclo de vida (nascimento, manutenção e destruição);
- Disponibilidade - propriedade que garante que a informação esteja sempre disponível para o uso legítimo, ou seja, por aqueles utilizadores autorizados pelo proprietário da informação.

O nível de segurança desejado, pode se consubstanciar em uma "política de segurança" que é seguida pela organização ou pessoa, para garantir que uma vez estabelecidos os princípios, aquele nível desejado seja perseguido e mantido.

Para estabelecer esta política, deve-se levar em conta:

- Riscos associados à falta de segurança;
- Benefícios;
- Custos de implementação dos mecanismos.

2.6.2 Mecanismos de segurança

O suporte para as recomendações de segurança pode ser encontrado em (FERREIRA, 2013):

- Controlos físicos: são barreiras que limitam o contato ou acesso direto a informação ou a infraestrutura (que garante a existência da informação) que a suporta.

Existem mecanismos de segurança que apoiam os controlos físicos, como portas, trancas, paredes, blindagem, guardas, entre outros.

- Controlos lógicos: são barreiras que impedem ou limitam o acesso a informação, que está em ambiente controlado, geralmente eletrónico, e que, de outro modo, ficaria exposta a alteração não autorizada por elemento mal-intencionado.

Existem mecanismos de segurança que apoiam os controlos lógicos:

- ✓ Mecanismos de criptografia. Permitem a transformação reversível da informação de forma a torná-la ininteligível a terceiros. Utiliza-se para tal, algoritmos determinados e uma chave secreta para, a partir de um conjunto de dados não criptografados, produzir uma sequência de dados criptografados. A operação inversa é a decifração;

Para garantir a segurança das informações foi utilizado o MD5 que é uma técnica utilizada na criptografia. Foi implementada uma classe que criptografa os dados de senhas usadas na autenticação (login).

O MD5 (Message-Digest Algorithm 5) ¹³ é um algoritmo de *hash* de 128 bits unidirecional desenvolvido pela RSA Data Security, e muito utilizado por *softwares* com protocolo ponto-a-ponto (P2P), verificação de integridade e *logins*.

Foi desenvolvido em 1991 por Ronald Rivest para suceder ao MD4 que tinha alguns problemas de segurança. Por ser um algoritmo unidirecional, uma hash md5 não pode ser transformada novamente no texto que lhe deu origem. O método de verificação é, então, feito pela comparação das duas hash (uma da base de dados, e a outra da tentativa de login). O MD5 também é usado para verificar a integridade de um ficheiro, através por exemplo, do programa md5sum, que cria a hash de um ficheiro. Isto pode-se tornar muito útil para *downloads* de ficheiros grandes, para programas P2P que constroem o ficheiro através de pedaços e estão sujeitos à corrupção dos mesmos.

O acesso ao sistema procura proteger o utilizador perante possíveis acessos não autorizados. Ao ter acesso ao sistema com o utilizador e a senha, automaticamente o sistema faz o uso do MD5 para garantir que sistemas maliciosos não capturem a mesma, caso ocorra não terão acesso ao sistema, pois para terem acesso terão que ter a senha que gerou e não ela criptografada.

Exemplo de uma senha gerada pelo MD5:

Senha sem MD5:123456

Senha com MD5: e10adc3949ba59abbe56e057f20f883e

- ✓ Assinatura digital. Um conjunto de dados criptografados, associados a um documento do qual são função, garantindo a integridade do documento associado, mas não a sua confidencialidade;

¹³<http://www.devmedia.com.br/criptografia-md5/2944> Acesso em 28/02/2014 às 18:37

- ✓ Mecanismos de garantia da integridade da informação. Usando funções de "Hashing" ou de controlo, consistindo na adição;
- ✓ Mecanismos de controlo de acesso. Palavras-chave, sistemas biométricos, firewalls, cartões inteligentes;

Para garantir a segurança nas transações, foi feita a gestão de utilizadores e foi definido os seus diferentes níveis de acesso.

A figura 7 mostra o modelo entidade-relacionamento relativo à gestão de utilizadores, em que é possível definir as suas diferentes transações e os diferentes menus a que têm acesso.

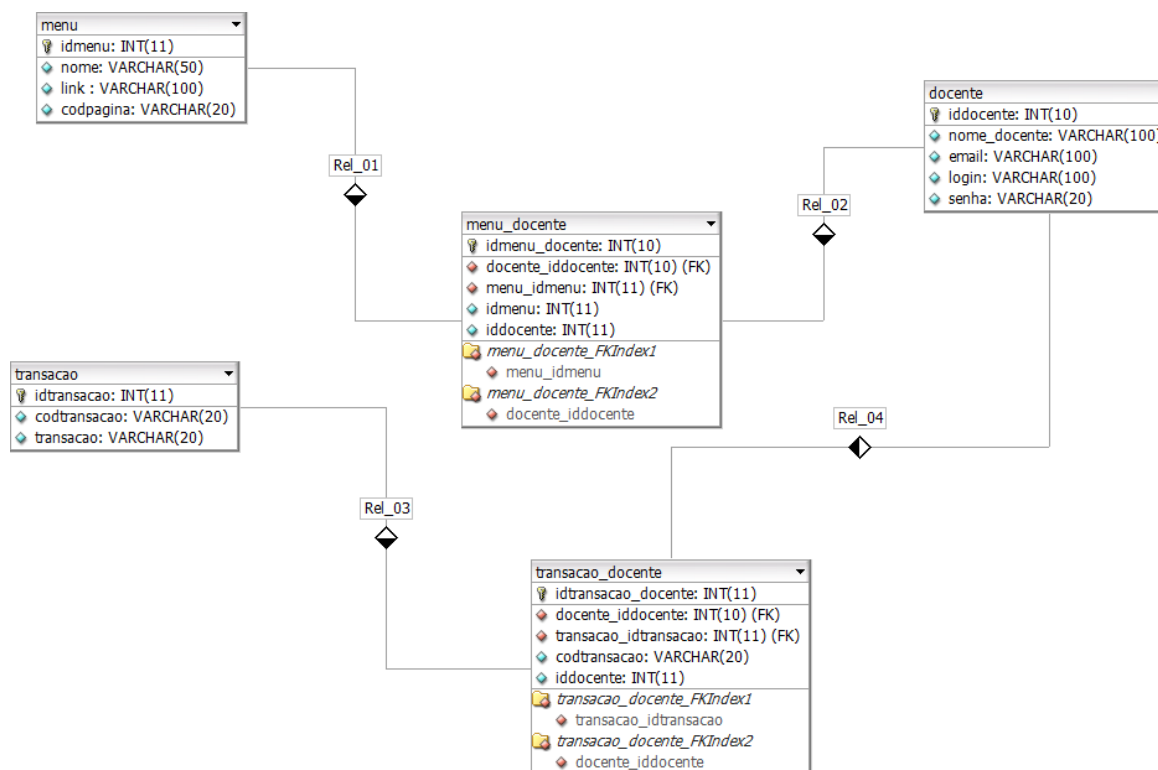


Figura 7 - Mer gestão de utilizadores

- ✓ Mecanismos de certificação. Atesta a validade de um documento;
- ✓ Integridade. Medida em que um serviço/informação é genuíno, isto é, está protegido contra a personificação por intrusos;
- ✓ Honeypot: É o nome dado a um *software*, cuja função é detetar ou de impedir a ação de um cracker, de um spammer, ou de qualquer agente externo estranho ao sistema, enganando-o, fazendo-o pensar que esteja de fato explorando uma vulnerabilidade daquele sistema;
- ✓ Protocolos seguros: uso de protocolos que garantem um grau de segurança e usam alguns dos mecanismos citados aqui. Existe hoje em dia um elevado número de ferramentas e sistemas que pretendem fornecer segurança. Alguns exemplos são os detentores de intrusões, os anti-vírus, firewalls, firewalls locais, filtros anti-spam, fuzzers, analisadores de código, etc.

2.6.3 Ameaças à segurança

As ameaças à segurança da informação são relacionadas diretamente à perda de uma de suas 3 características principais, quais sejam:

- Perda de Confidencialidade: seria quando há uma quebra de sigilo de uma determinada informação (ex: a senha de um utilizador ou administrador de sistema) permitindo com que sejam expostas informações restritas as quais seriam acessíveis apenas por um determinado grupo de utilizador;
- Perda de Integridade: aconteceria quando uma determinada informação fica exposta a manuseio por uma pessoa não autorizada, que efetua alterações que não foram aprovadas e não estão sob o controlo do proprietário (corporativo ou privado) da informação;

- Perda de Disponibilidade: acontece quando a informação deixa de estar acessível por quem necessita dela. Seria o caso da perda de comunicação com um sistema importante para a empresa, que aconteceu com a queda de um servidor de uma aplicação crítica de negócio, que apresentou uma falha devido a um erro causado por motivo interno ou externo ao equipamento. No caso de ameaças à rede de computadores ou a um sistema, estas podem vir de agentes maliciosos, muitas vezes conhecidos como crackers, (hackers não são agentes maliciosos, pois tentam ajudar a encontrar possíveis falhas). Estas pessoas são motivadas para fazer esta ilegalidade por vários motivos. Os principais são: notoriedade, autoestima, vingança e o dinheiro. De acordo com pesquisa elaborada pelo Computer Security Institute, mais de 70% dos ataques partem de usuários legítimos de sistemas de informação (Insiders) - o que motiva corporações a investir largamente em controlos de segurança para seus ambientes corporativos (intranet).

2.6.4 Nível de segurança

Depois de identificado o potencial de ataque, as organizações têm que decidir o nível de segurança a estabelecer para uma rede ou sistema os recursos físicos e lógicos a necessitar de proteção. No nível de segurança devem ser quantificados os custos associados aos ataques e os associados à implementação de mecanismos de proteção para minimizar a probabilidade de ocorrência de um ataque (FERREIRA, 2013).

- Segurança física

Considera as ameaças físicas como incêndios, desabamentos, relâmpagos, alagamento, acesso indevido de pessoas, forma inadequada de tratamento e manuseamento do material.

- Segurança lógica

Atenta contra ameaças ocasionadas por vírus, acessos remotos à rede, backup desatualizados, violação de senhas, etc.

2.6.5 Políticas de segurança

De acordo com o RFC 2196 (The Site Security Handbook), uma política de segurança consiste num conjunto formal de regras que devem ser seguidas pelos utilizadores dos recursos de uma organização.

As políticas de segurança devem ter implementação realista, e definir claramente as áreas de responsabilidade dos utilizadores, do pessoal de gestão de sistemas e redes e da direção. Deve também adaptar-se a alterações na organização.

As políticas de segurança fornecem um enquadramento para a implementação de mecanismos de segurança, definem procedimentos de segurança adequados, processos de auditoria à segurança e estabelecem uma base para procedimentos legais na sequência de ataques.

O documento que define a política de segurança deve deixar de fora todos os aspetos técnicos de implementação dos mecanismos de segurança, pois essa implementação pode variar ao longo do tempo. Deve ser também um documento de fácil leitura e compreensão, além de resumido.

Algumas normas definem aspetos que devem ser levados em consideração ao elaborar políticas de segurança. Entre essas normas estão a BS 7799 (elaborada pela British Standards Institution) e a NBR ISO/IEC 17799 (a versão brasileira desta primeira). A ISO começou a publicar a série de normas 27000, em substituição à ISO 17799 (e por conseguinte à BS 7799), das quais a primeira, ISO 27001, foi publicada em 2005.

Existem duas filosofias por trás de qualquer política de segurança: a proibitiva (tudo que não é expressamente permitido é proibido) e a permissiva (tudo que não é proibido é permitido).

Os elementos da política de segurança devem ser considerados:

- A Disponibilidade: o sistema deve estar disponível de forma que quando o usuário necessitar possa usar. Dados críticos devem estar disponíveis ininterruptamente;
- A Utilização: o sistema deve ser utilizado apenas para os determinados objetivos;
- A Integridade: o sistema deve estar sempre íntegro e em condições de ser usado;
- A Autenticidade: o sistema deve ter condições de verificar a identidade dos usuários, e este ter condições de analisar a identidade do sistema;
- A Confidencialidade: dados privados devem ser apresentados somente aos donos dos dados ou ao grupo por ele liberado.

Políticas de Senhas

Dentre as políticas utilizadas pelas grandes corporações a composição da senha ou password é a mais controversa. Por um lado profissionais com dificuldade de memorizar várias senhas de acesso, por outros funcionários displicentes que anotam a senha sob o teclado no fundo das gavetas, em casos mais graves o colaborador anota a senha no monitor.

Recomenda-se a adoção das seguintes regras para minimizar o problema, mas a regra fundamental é a conscientização dos colaboradores quanto ao uso e manutenção das senhas.

- Senha com data para expiração

Adota-se um padrão definido onde a senha possui prazo de validade com 30 ou 45 dias, obrigando o colaborador ou utilizador a renovar sua senha;

10 ANOS DE EMPENHO E DEDICAÇÃO

- Inibir a repetição

Adota-se através de regras predefinidas que uma senha uma vez utilizada não poderá ter mais que 60% dos caracteres repetidos, p. ex: senha anterior “123senha” nova senha deve ter 60% dos caracteres diferentes como “456seuze”, neste caso foram repetidos somente os caracteres “s” “e” os demais diferentes;

- Obrigar a composição com número mínimo de caracteres numéricos e alfabéticos

Define-se obrigatoriedade de 4 caracteres alfabéticos e 4 caracteres numéricos, por exemplo: 1s4e3u2s ou posicionais os 4 primeiros caracteres devem ser numéricos e os 4 subsequentes alfabéticos por exemplo: 1432seuz;

- Criar um conjunto possíveis senhas que não podem ser utilizadas

Monta-se uma base de dados com formatos conhecidos de senhas e proibir o seu uso, como por exemplo o usuário chama-se José da Silva, logo sua senha não deve conter partes do nome como 1221jose ou 1212silv etc, os formatos DDMMAAAA ou 19XX, 1883emc ou I2B3M4 etc.

CAPÍTULO 3 – CARACTERIZAÇÃO DA UNIVERSIDADE DO MINDELO

3.1 Dados Gerais

A Universidade do Mindelo é fruto do sucesso e prestígio conseguidos, como forma de responder aos novos desafios que Cabo Verde enfrenta.

A Universidade do Mindelo pretende valorizar e promover a intervenção do sector do ensino privado no sistema educativo, como forma de diversificar as possibilidades de acesso de todos os cabo-verdianos à Educação-Formação. Pretende servir uma população juvenil diversificada, desde os mais jovens, os que terminaram recentemente o 12º ano do Ensino Secundário/Ano Zero, aos mais velhos, como sejam aqueles que já saíram do sistema educativo há mais tempo.

A Universidade encontra-se estruturada, em Departamentos, cujos responsáveis fazem parte de um conselho científico. A gestão é confiada a um Conselho Diretivo composta pelo Reitor, pelo Vice-Reitor, pelo Presidente do Conselho Científico, pelo Presidente do Conselho Pedagógico, pelo Presidente dos Serviços Académicos e Administrativos e pelo Presidente da Associação de Estudantes.

A Reitoria é representada pelo Reitor Albertino Emanuel Lopes Graça, que é o órgão executivo da administração superior que coordena e supervisiona todas as atividades da Instituição Universitária.

3.2 Visão

“Servir com qualidade e organização”

Fazer com que os Serviços Académicos e Administrativos sejam um serviço de referência ao nível da comunidade académica da UM suscetível de prestar serviços qualificados, fazendo uso de processos internos modernos, assente no uso intensivo das tecnologias de informação e comunicação, racionalizando procedimentos, criando condições de trabalho adequadas ao reforço das competências dos seus profissionais, tudo em vista da obtenção de uma maior qualidade dos serviços prestados tal como percecionada pelos seus utentes, e da realização profissional dos seus colaboradores.

3.3 Missão

“Transformar os sonhos dos nossos alunos em realidade, através da educação e encaminhamento ao mercado de trabalho.”

3.4 Valores

- Orientação para o utente – Esforço no sentido de conhecer as necessidades dos nossos utentes;
- Transparência – procura, no respeito por lei e pelos direitos e interesses legalmente protegidos, dar a conhecer os termos e resultados de trabalho, dos regulamentos (Regulamento Interno e Manual de Acolhimento dos Docentes para os SAA e das Contas e Posicionamento Financeiro para os SCF);
- Humildade – Um serviço que aprende permanentemente, introduzindo as alterações necessárias à melhoria da qualidade dos serviços prestados;
- Profissionalismo – O pessoal da SAA procura dotar-se das qualificações necessárias para o exercício das suas tarefas e procura realizá-las com eficiência, eficácia, qualidade, no sentido de promover uma forte empatia com os seus utentes;

- Compromisso com a qualidade – a gestão visa a melhoria contínua dos serviços prestados, e tem como fim último a excelência do seu desempenho.

3.5 Estrutura Organizacional

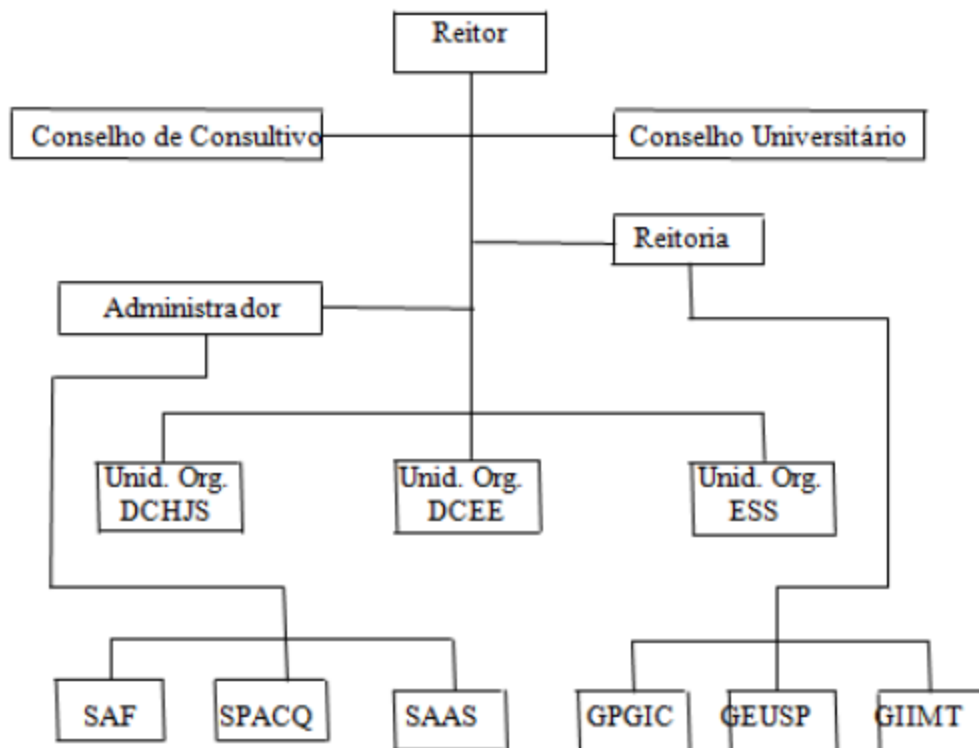


Figura 8 - Organograma da UM

DCHJS – Departamento das Ciências Humanas, Jurídicas e Sociais

DCEE - Departamento de Ciências Económicas e Empresariais

ESS – Escola Superior de Saúde

SAF – Serviços Académicos e Administrativos

SPACQ – Serviços Planeamento, Auditoria e Controlo de Qualidade

SAAS – Serviços Académicos e de Acção Social

CAPÍTULO 4 – DESENVOLVIMENTO DO PROTÓTIPO

Este capítulo tem por objetivo demonstrar as fases executadas para o levantamento de requisitos, análise e desenvolvimento do sistema, com a seguinte ordem de apresentação: análise dos requisitos: contemplando os requisitos funcionais e não funcionais, a especificação contendo os principais casos de uso, modelação dos dados, seguidos pela implementação, operacionalidade da implementação e resultados obtidos.

4.1 Requisitos principais do sistema

Um requisito num sistema é uma funcionalidade ou característica considerada relevante na ótica do utilizador. Normalmente, representa o comportamento esperado do sistema, que na prática consiste num serviço que deve ser disponibilizado a um utilizador (Nunes e O'Neill, 2004).

Os requisitos podem ser classificados em 2 categorias:

- **Requisitos funcionais** – descrevem o que o sistema faz ou é esperado que faça. Estes são os requisitos que inicialmente serão levantados, abrangendo a descrição de processamentos a efetuar pelo sistema, entrada (inputs), e saídas (outputs) de informação em papel ou no ecrã que derivam da interação com pessoas e outros sistemas.
- **Requisitos não funcionais** – estão relacionados com as características qualitativas do sistema, descrevendo a qualidade com que o sistema deverá fornecer os requisitos funcionais.

O levantamento de requisitos foi definido através de entrevista realizada com o responsável pelo processo de alocação de salas e com o Coordenador do Curso de Informática de Gestão, a

fim de recolher as informações sobre as necessidades para o sistema. A referida entrevista encontra-se no apêndice A deste relatório.

Para contemplar os objetivos do trabalho, enumerados no capítulo 1, levantou-se os requisitos funcionais e não funcionais, apresentados a seguir:

4.1.1 Requisitos funcionais

ID	Funcionalidade	Descrição	Actor
R1	Cadastro utilizador, perfis e acessos	Efetuar o cadastro do utilizador no sistema	Administrador
R2	Autenticar utilizador	O sistema deve autenticar o utilizador no sistema mediante a informação de login e senha.	Administrador, docente e gestor de salas.
R3	Cadastro sala	Efetuar o cadastro de salas	Gestor de salas
R4	Cadastro docente	Efetuar o cadastro de dados do docente	Gestor de salas
R5	Cadastro disciplina	Cadastro de dados da disciplina	Gestor de salas
R6	Cadastro curso	Cadastro de dados do curso	Gestor de salas
R7	Cadastro horário	Cadastro de dados dos horários	Gestor de salas
R8	Consulta reserva	Consulta as reservas	Gestor de salas, docente
R9	Efetua reserva	Efetua o registo de reserva no sistema	Gestor de salas e docente
R10	Emitir relatório	Emitir relatórios	Gestor de salas
R11	Controlo de acessos	O sistema de vera permitir o controlo de acessos	Administrador
R12	Gestão de utilizadores	O sistema deve permitir adicionar e remover utilizadores	Administrador

Tabela 1 - Requisitos funcionais

4.1.2 Requisitos não funcionais:

Requisitos	Descrição
<ul style="list-style-type: none"> Navegadores 	Os navegadores podem ser <i>Mozilla Firefox</i> ou <i>Internet Explorer</i> .
<ul style="list-style-type: none"> Desenvolvimento 	Para o desenvolvimento do sistema será utilizada a linguagem <i>PHP</i> e <i>HTML</i> e a tecnologia <i>Javascript</i>
<ul style="list-style-type: none"> Servidor 	O servidor web a ser utilizado será o <i>Apache</i>
<ul style="list-style-type: none"> Base de dados 	O sistema deverá fazer o uso da base de dados <i>MySQL</i> .

Tabela 2 - Requisitos não funcionais 1

4.2 Lista dos Casos de Uso

Um caso de uso é um documento que descreve a sequência de eventos de um ator (um agente externo) que usa um sistema para completar um processo. Casos de uso não são exatamente especificação de requisitos ou especificação funcional, porém ilustram e implicam requisitos na história que eles contam.

Para o sistema de reserva de salas, foram identificados os seguintes casos de uso que encontram-se discriminados na tabela 3 a seguir:

Casos de uso	Descrição
Controle de acesso	Permite dar acesso aos diferentes utilizadores do sistema de acordo com o perfil que possuir.
Gestão de utilizadores	Permite identificar os utilizadores do sistema e fazer a sua separação em grupos.
Cadastro docente	Permite adicionar o docente ao sistema
Cadastro sala	Permite adicionar a sala ao sistema
Cadastro horário	Permite adicionar o horário ao sistema
Cadastro disciplina	Permite adicionar a disciplina ao sistema

10 ANOS DE EMPENHO E DEDICAÇÃO

Cadastro curso	Permite adicionar o curso ao sistema
Efetuar Reserva	Regista as reservas efetuadas
Consultar Reservas	Consultar as reservas que foram efetuadas
Emitir relatórios	Emissão de relatórios com as informações da base de dados do sistema.

Tabela 3 - Lista de casos de uso

4.3 Análise do Sistema

Especificou-se o sistema com análise orientada a objetos utilizando a *Unified Modeling Language* (UML). Utilizou-se o ArgoUML para elaborar os diagramas de casos de uso, de classe e de sequência. Para a modelação dos dados utilizou-se o MySQL Workbench.

4.3.1 Diagrama de casos de uso

Esta secção apresenta os diagramas de caso de uso dos diferentes perfis do sistema desenvolvido, sendo os diagramas gerados através da ferramenta *Microsoft Office Visio*. Para o melhor entendimento do projeto, os detalhes dos casos de uso se encontram no apêndice B.

Primeiramente foram identificados os atores que interagem com o sistema. Sendo eles:

- **Administrador:** é o utilizador com maior privilégio dentro do sistema;
- **Gestor de salas:** é o utilizador que possui perfil de acesso para fazer a manutenção dos registos da Universidade;
- **Docente:** é o utilizador responsável pela alteração dos seus dados e das suas reservas.

Após a identificação dos atores, foram identificados para cada ator os casos de utilização em que este interage com o sistema. Estes encontram-se na tabela 4 a seguir:

10 ANOS DE EMPENHO E DEDICAÇÃO

Actor	Casos de Utilização
Docente	<ul style="list-style-type: none"> • Efetuar Reserva • Consultar Reservas
Gestor de Salas	<ul style="list-style-type: none"> • Cadastro de Docentes • Cadastro de Salas • Cadastro de cursos • Cadastro de disciplinas • Cadastro de horários • Cadastro de turma • Confirmar reservas • Gerar relatórios
Administrador	<ul style="list-style-type: none"> • Gestão de utilizadores • Controlo de acesso

Tabela 4 - Casos de uso do sistema

Diagrama de caso de uso geral do sistema

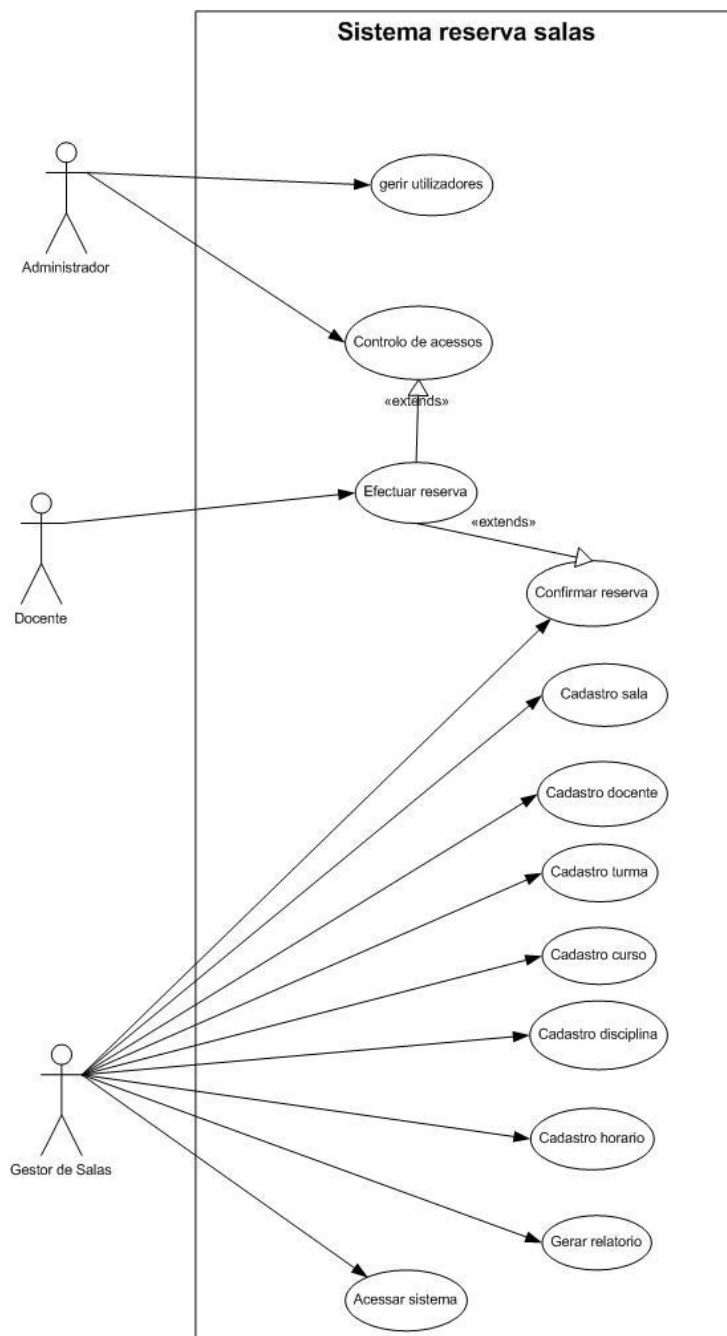


Figura 9 - Diagrama de caso uso geral do sistema

Diagrama de casos de uso para o administrador

O Administrador, que é o utilizador que possui total controlo das configurações do sistema. A figura 10 apresenta o diagrama de casos de uso para o administrador do sistema.

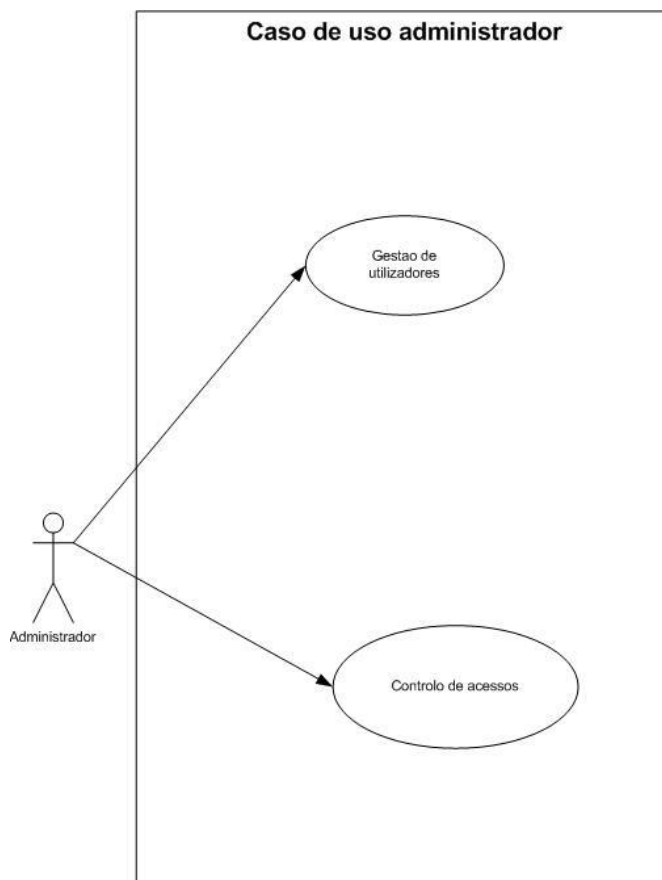


Figura 10 - Diagrama de caso de uso administrador do sistema

Diagrama de casos de uso para o Gestor de salas

A figura 11 apresenta o diagrama de casos de uso com os casos de uso do Gestor de salas, que é o responsável pela manutenção dos registos no âmbito da Universidade.

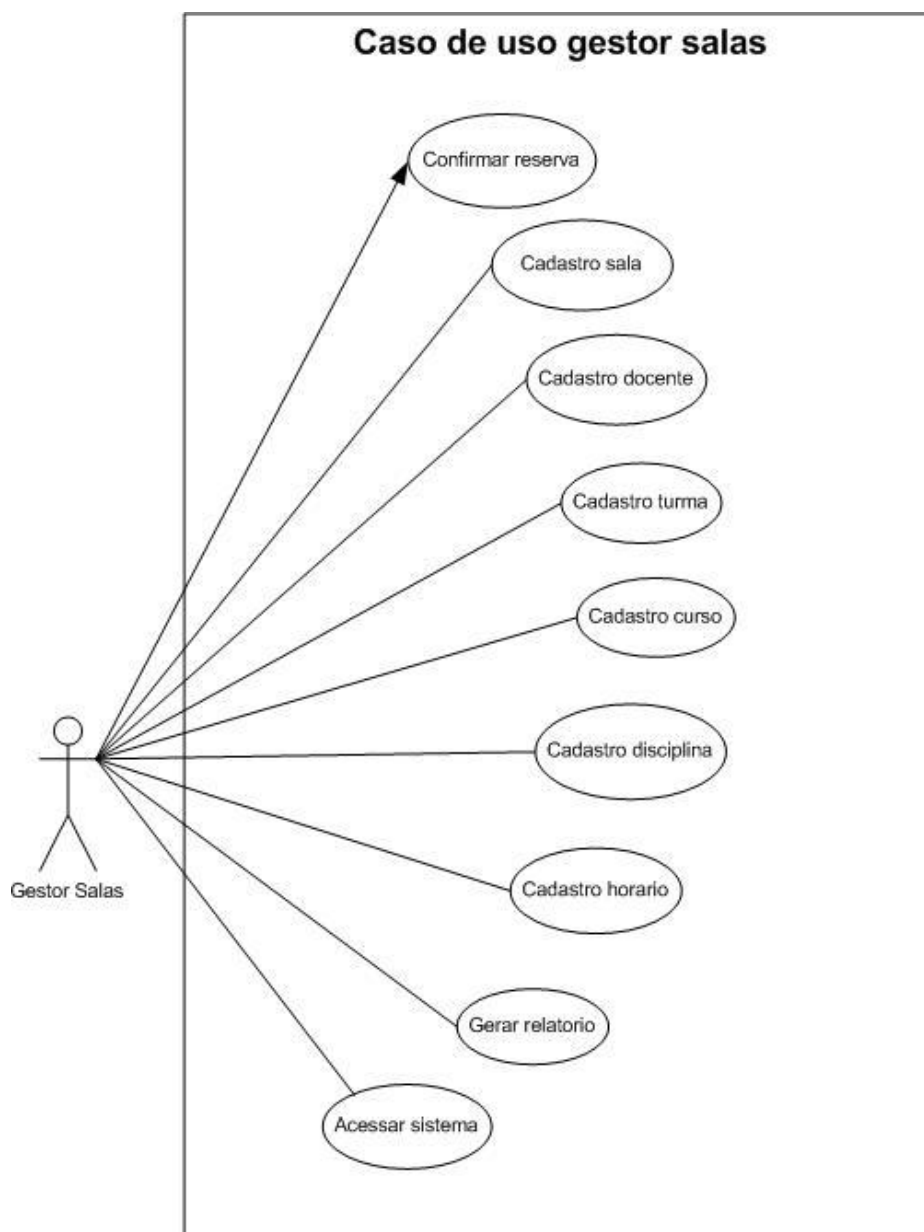


Figura 11-Diagrama caso uso gestor salas

Diagrama de casos de uso para o Docente

A figura 12 apresenta o diagrama de casos de uso para o ator Docente.

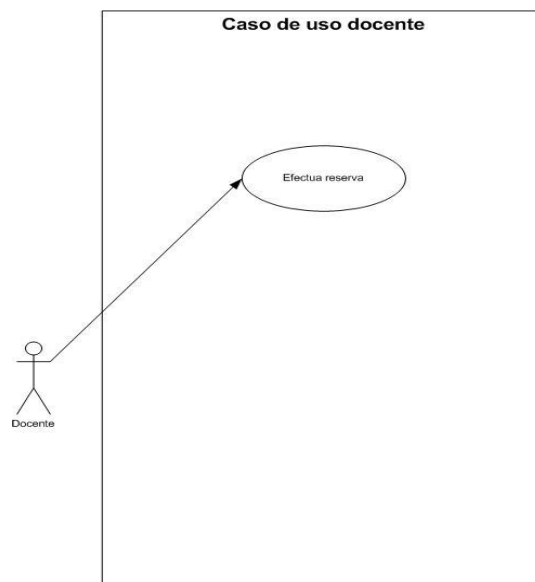


Figura 12 - Diagrama caso uso docente 1

4.3.2 Diagramas de sequência

Caso de uso: Efetuar reserva

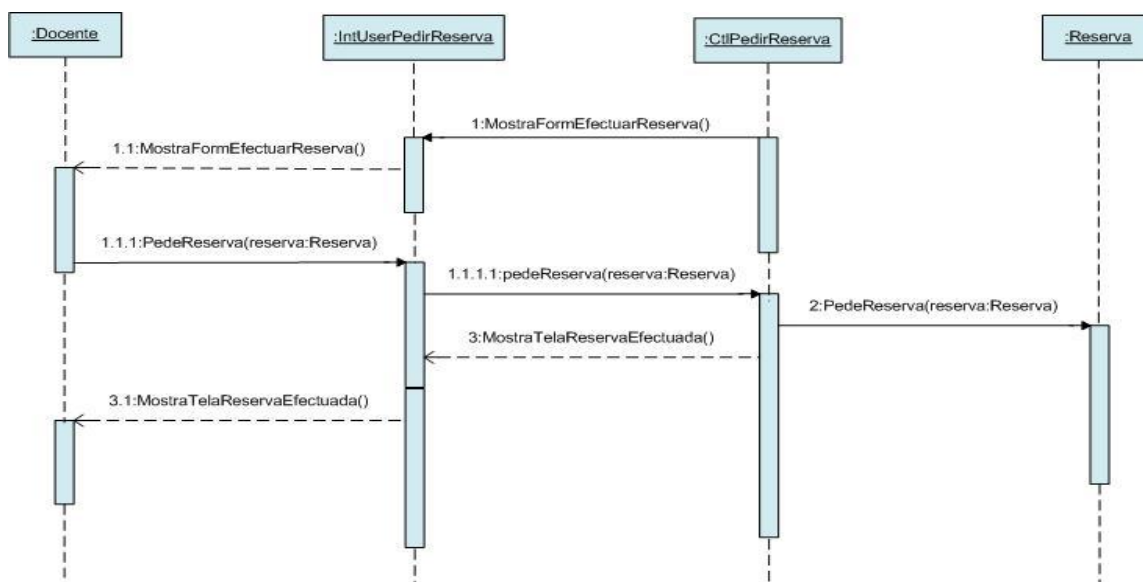


Figura 13 - Diagrama sequencia efetuar reserva

Caso de uso: Cadastro sala

10 ANOS DE EMPENHO E DEDICAÇÃO

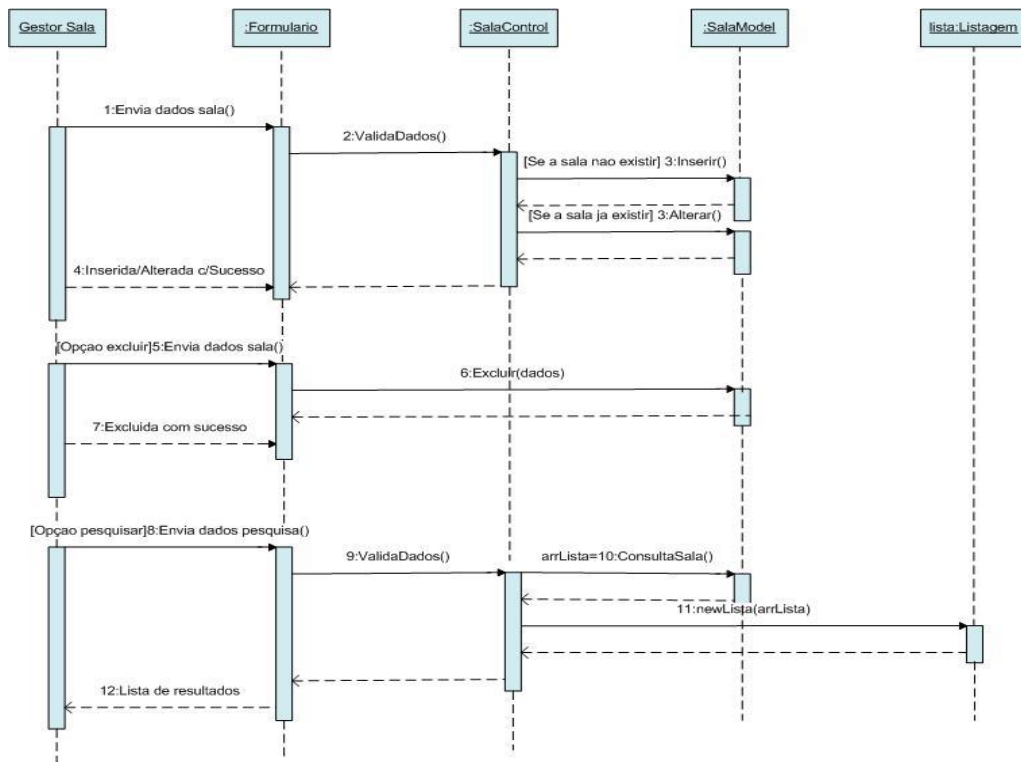


Figura 14 - Diagrama sequencia cadastro sala

Caso de uso: Cadastro docente

10 ANOS DE EMPENHO E DEDICAÇÃO

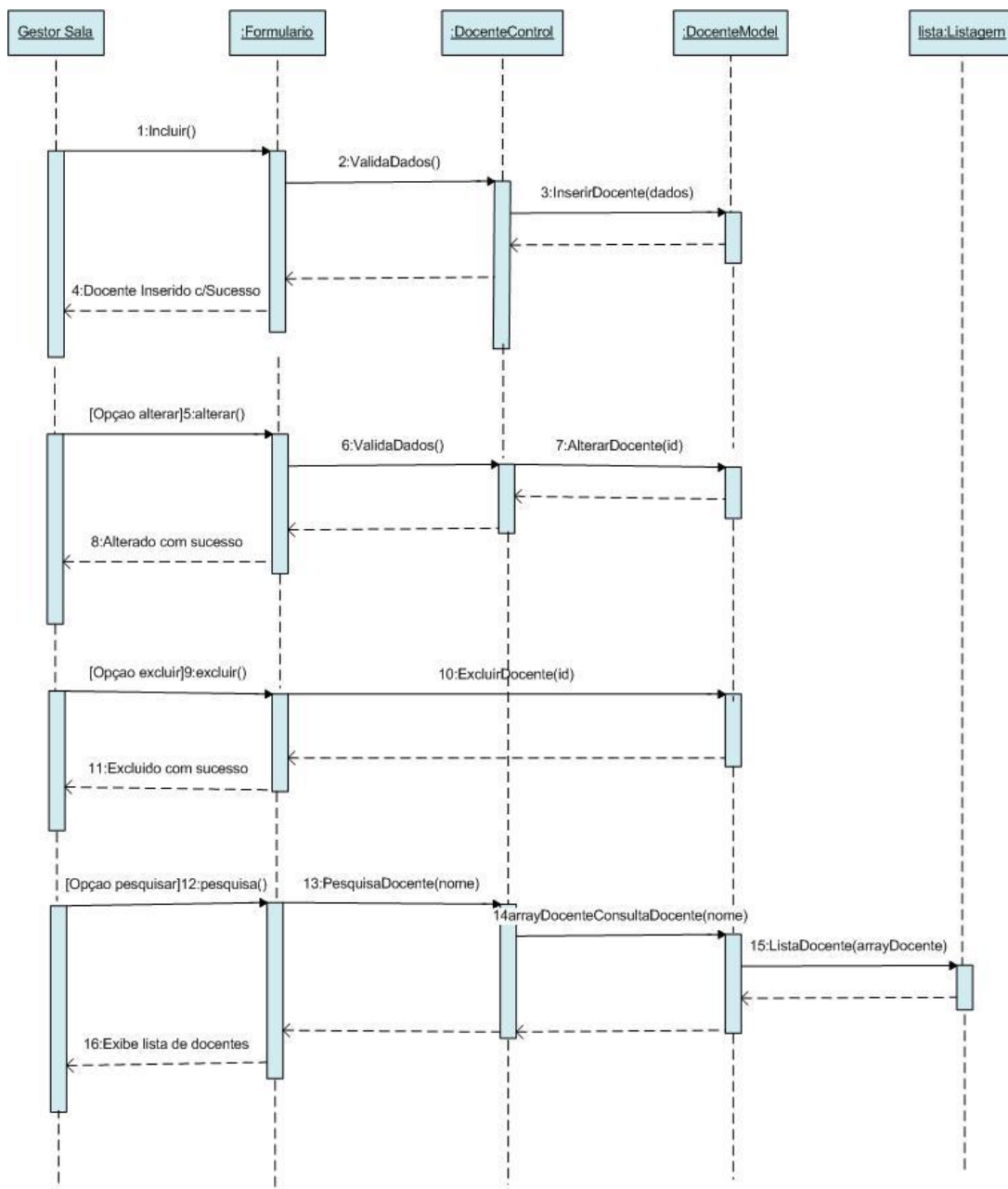


Figura 15-Diagrama sequencia cadastro docente

Caso de uso: Gerar relatório

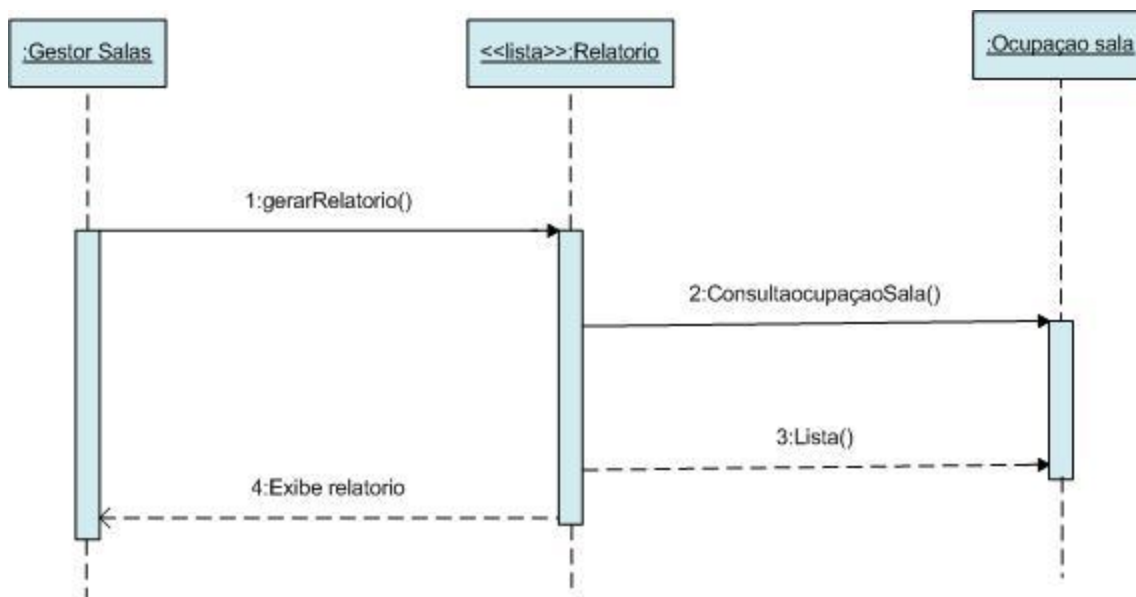


Figura 16 - Diagrama sequencia gerar relatorio

4.4 Projeto do Sistema

No projeto deste sistema trabalhamos a modelagem dos dados que nos permite uma melhor compreensão da arquitetura e do comportamento do que será implementado. Independentemente do grau de complexidade da aplicação, a correta modelagem dos seus dados a torna mais robusta e fácil de manter. A modelagem dos dados nos apresenta as funcionalidades do sistema.

Uma base de dados relacional foi utilizada para armazenar todos os dados manipulados pelo sistema proposto. Neste subcapítulo apresentamos o projeto da base de dados.

4.4.1 Projeto da base de dados

Neste trabalho, o projeto de base de dados apresenta o esquema conceitual para a base de dados utilizando-se o modelo de dados Entidade-Relacionamento (ER). Este esquema conceitual apresenta uma descrição concisa dos requisitos de dados.

4.4.1.1 Modelo entidade relacionamento (MER)

O modelo de dados compreende as tabelas para o cadastro das entidades envolvidas, que são: curso, docente, disciplina, horário, sala e reserva.

A figura 17 a seguir representa o MER desenvolvido para o sistema, que foi implementado com o auxílio da ferramenta DBDesigner4.

10 ANOS DE EMPENHO E DEDICAÇÃO

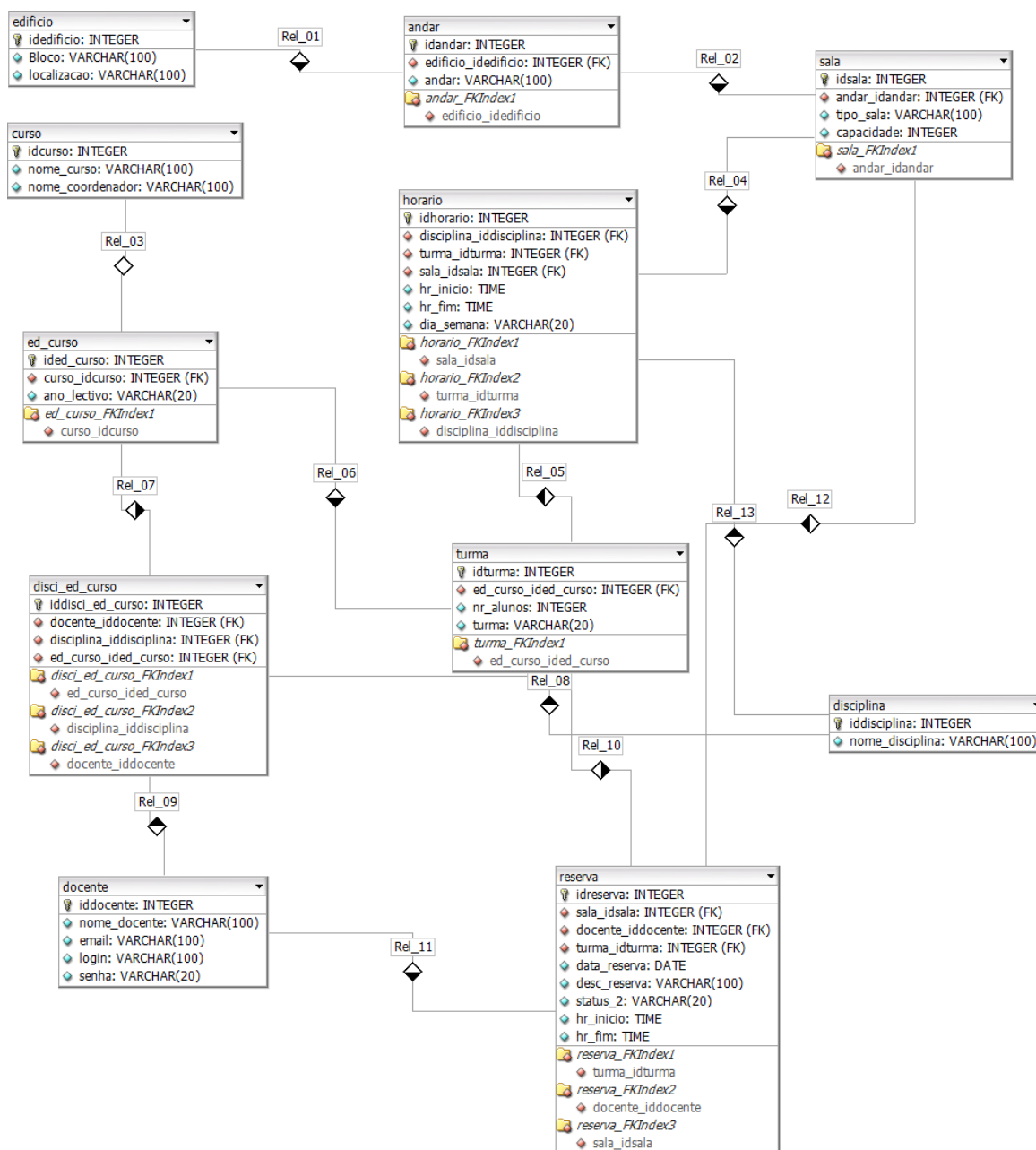


Figura 17 - Diagrama Entidade Relacionamento

O dicionário de dados foi utilizado neste trabalho para descrever o significado das tabelas do sistema e os atributos das tabelas relacionadas.

Abaixo seguem as tabelas referentes ao detalhamento das entidades persistentes na base de dados:

Tabela edificio			
Nome	Tipo de dados	Tamanho	Descrição
idedificio	Integer	11	Identificador do edifício. Chave primária
Bloco	varchar	100	Bloco do edifício.
Localização	varchar	100	Localização do edifício

Tabela 5 - Dicionário dados tabela edifício

Tabela andar			
Nome	Tipo de dados	Tamanho	Descrição
idandar	Int	11	Identificador do andar. Chave primária
Andar	Varchar	100	Nome do andar.
idedificio	Integer	11	Identificador do edificio. Chave estrangeira da tabela edificio

Tabela 6 - Dicionário dados tabela andar

Tabela sala			
Nome	Tipo de dados	Tamanho	Descrição
idsala	Integer	11	Código da sala
tipo_sala	Varchar	100	Tipo da sala.chave primaria
Capacidade	Integer	11	Capacidade da sala
idandar	Integer	11	Andar da sala. Chave estrangeira da tabela andar.

Tabela 7 - Dicionário dados tabela sala

Tabela curso			
Nome	Tipo de dados	Tamanho	Descrição
idcurso	Integer	11	Código do curso.Chave primária
nome_curso	Varchar	100	Nome do curso.
nome_coordenador	Varchar	100	Nome do coordenador do curso.

10 ANOS DE EMPENHO E DEDICAÇÃO

Tabela 8 - Dicionário dados tabela curso

Tabela disciplina			
Nome	Tipo de dados	Tamanho	Descrição
iddisciplina	Integer	11	Código da disciplina. Chave primária
nome_disciplina	Varchar	100	Nome da disciplina.

Tabela 9 - Dicionário dados tabela disciplina

Tabela docente			
Nome	Tipo de dados	Tamanho	Descrição
iddocente	Integer	11	Código do docente. Chave primária
Nome_docente	Varchar	100	Nome do docente.
Email	Varchar	100	Email
Login	Varchar	100	Login
Senha	Varchar	50	Senha

Tabela 10 - Dicionário dados tabela docente

Tabela horario			
Nome	Tipo de dados	Tamanho	Descrição
idhorario	Integer	11	Identificador do horario. Chave primária
hora_inicio	Time		Hora início
hora_fim	Time		Hora fim.
Dia_semana	Varchar	20	numero do semestre
iddisciplina	Integer	11	Id disciplina. Chave estrangeira da tabela Disciplina
idturma	Integer	11	Id turma. Chave estrangeira da tabela turma.
idsala	Integer	11	Id da sala. Chave estrangeira da tabela sala

Tabela 11 - Dicionário dados tabela horario

Tabela reserva			
Nome	Tipo de dados	Tamanho	Descrição
idreserva	Integer	11	Código da reserva. Auto-increment. Chave primária
hora_inicio	Time		Hora início. Chave estrangeira da tabela horário

10 ANOS DE EMPENHO E DEDICAÇÃO

hora_fim	Time		Hora fim. Chave estrangeira da tabela horário
idsala	Integer	11	id sala. Chave estrangeira da tabela sala
iddocente	Integer	11	iddocente. Chave estrangeira da tabela docente
idturma	Integer	11	Id turma. Chave estrangeira da tabela turma
Data_reserva	Date		Data da reserva
desc_reserva	Varchar	100	Descrição da reserva
Status_2	Varchar	20	Status da reserva

Tabela 12 - Dicionário dados tabela reserva

Tabela edição de curso			
Nome	Tipo de dados	Tamanho	Descrição
Ided_curso	Integer	11	Código da edição do curso. Chave primária
Ano_lectivo	Varchar	20	Ano lectivo.
idcurso	Integer	11	Id curso. Chave estrangeira da tabela curso.

Tabela 13 - Dicionário dados edição curso

Tabela Disciplina edição do curso			
Nome	Tipo de dados	Tamanho	Descrição
Iddisci_ed_curso	Integer	11	Código da disciplina edição curso. Chave primária
iddocente	Integer	11	Id docente. Chave estrangeira tabela docente
iddisciplina	Integer	11	Id disciplina. Chave estrangeira tabela disciplina
Ided_curso	Integer	11	Id edição curso. Chave estrangeira da tabela edição curso

Tabela 14 - Dicionário dados disciplina edição curso

Tabela Turma			
Nome	Tipo de dados	Tamanho	Descrição
Idturma	Integer	11	Código da turma. Chave primária
Nr_alunos	Integer	11	Número de alunos
turma	Varchar	20	Turma
Ided_curso	Integer	11	Id edição curso. Chave estrangeira da tabela edição curso

Tabela 15 - Dicionário dados turma

4.4.1.2 Diagrama de classes

Na figura 18, é apresentado o diagrama de classes elaborado para o Sistema. Neste diagrama são apresentadas as classes, seus atributos e métodos.

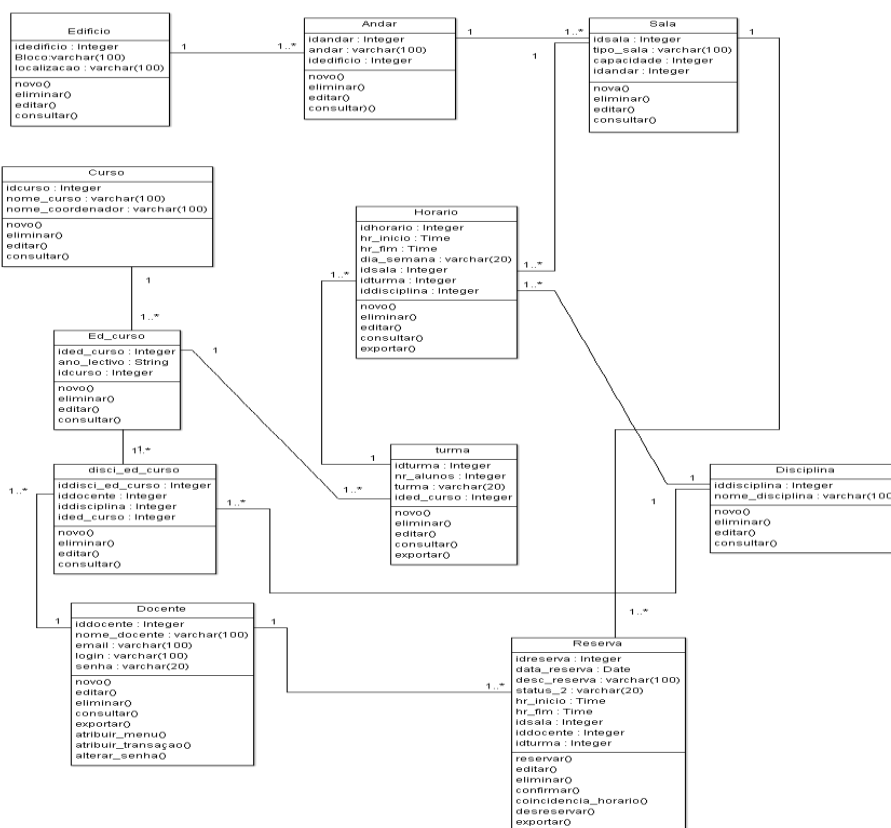


Figura 18 - Diagrama de classes

4.5 Implementação do Sistema

Esta secção demonstra os detalhes da implementação do projeto, bem como as técnicas e ferramentas utilizadas.

4.5.1 Técnicas e ferramentas utilizadas

Para implementação do sistema foi utilizada a linguagem de programação PHP. Para a validação de algumas informações foi utilizado o *Java Script*.

Para o desenvolvimento, foi utilizado o pacote Wamp, que inclui servidor Apache, o sistema de base de dados MySQL, o gerenciador de base de dados phpMyAdmin e o interpretador de páginas PHP.

4.5.2 Operacionalidade da implementação

Nesta subsecção é apresentada a sequência de telas e operações para cada tipo de utilizador, para conseguir utilizar corretamente o sistema para controlo e reserva de salas.

4.5.2.1 Interface de acesso ao sistema

Logo que o utilizador acede o sistema, ele encontra a página de login, como pode ser observado na figura 19. Nesta tela, o utilizador insere seu login e sua senha de acesso para poder aceder ao sistema.

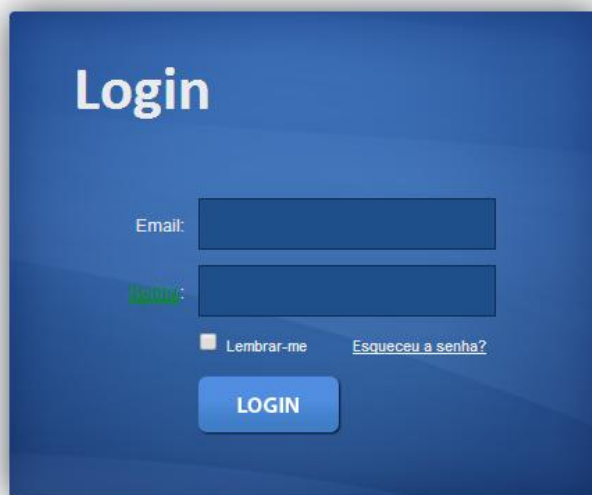


Figura 19 - Interface Acesso ao sistema

Caso as credenciais informadas estejam corretas, o utilizador acederá a página contendo um menu personalizado (de acordo com o seu nível de acesso). Na figura 20 é apresentada a tela da página que o utilizador “Administrador” (perfil de administrador) tem acesso.

4.5.2.2 Interface do Menu Principal

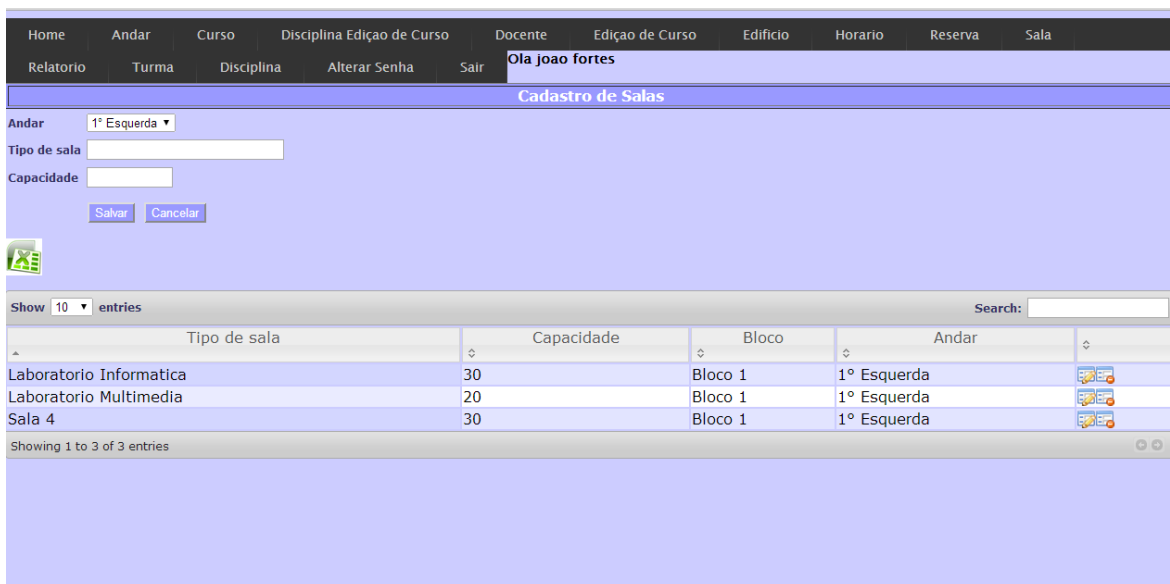


Figura 20 - Interface menu principal

Como ilustra a figura 20, na tela inicial do sistema há o menu. Dependendo do grupo de utilizador (Administrador, Gestor de salas ou Docente) o menu altera.

4.5.2.3 Cadastro de sala

A tela de manutenção das salas e suas características (figura 21) é onde são cadastrados, alterados ou excluídos andar, tipo de sala e capacidade.



Tipo de sala	Capacidade	Bloco	Andar
Laboratorio Informatica	30	Bloco 1	1º Esquerda
Laboratorio Multimedia	20	Bloco 1	1º Esquerda
Sala 4	30	Bloco 1	1º Esquerda

Figura 21 - Interface cadastro sala

4.5.2.4 Efetuar reserva

A figura 22 mostra a tela de reserva, esta tela é igual para todos os perfis de acesso e nesta é possível reservar uma sala para um determinado período.

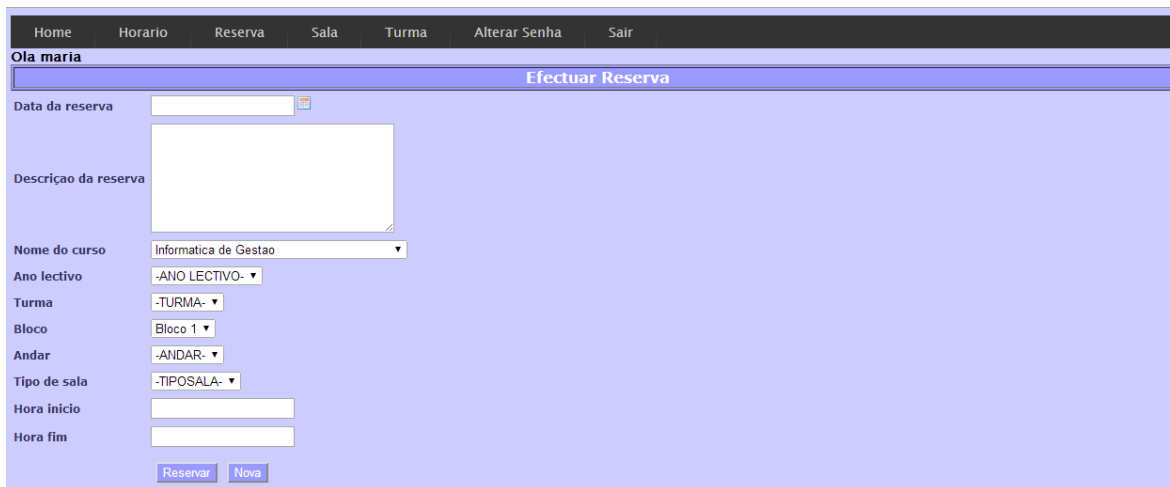


Figura 22- Interface efetuar reserva

Ao preencher os campos necessários e clicar sobre o botão “reservar” o sistema irá gerar uma locação.

4.5.2.5 Aceitar um pedido de reserva de um horário de uma sala

Nessa funcionalidade utilizadores do grupo Gestor de salas podem aceitar ou rejeitar pedidos de reservas.



Figura 23-Interface aceitar reserva 1

Para isto, basta pressionar o botão aceitar conforme ilustra a figura 23. Os docentes que pediram as reservas irão automaticamente receber um e-mail confirmando se a reserva foi aceita ou não.

4.5.2.6 Interface gestão de utilizadores

Nesta tela são solicitadas informações relacionadas ao cadastro de docente como: nome_docente, e-mail, login e senha.

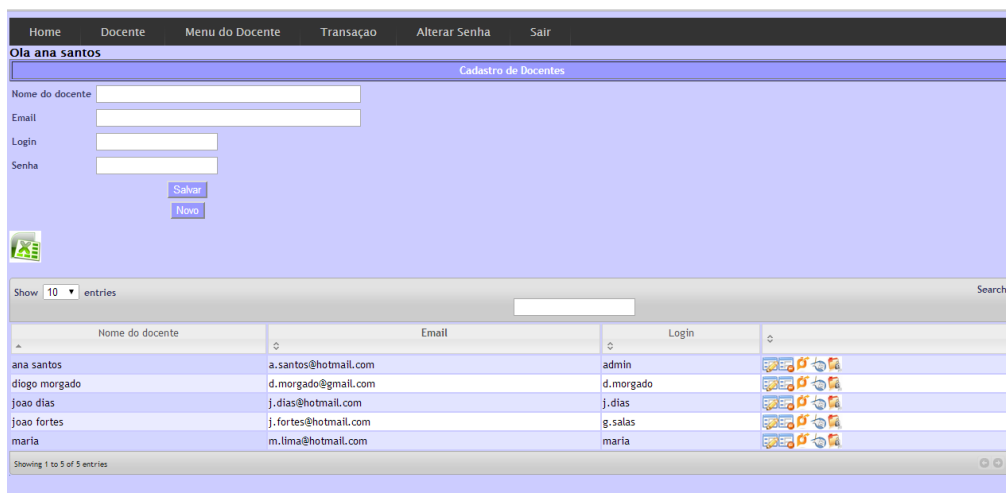
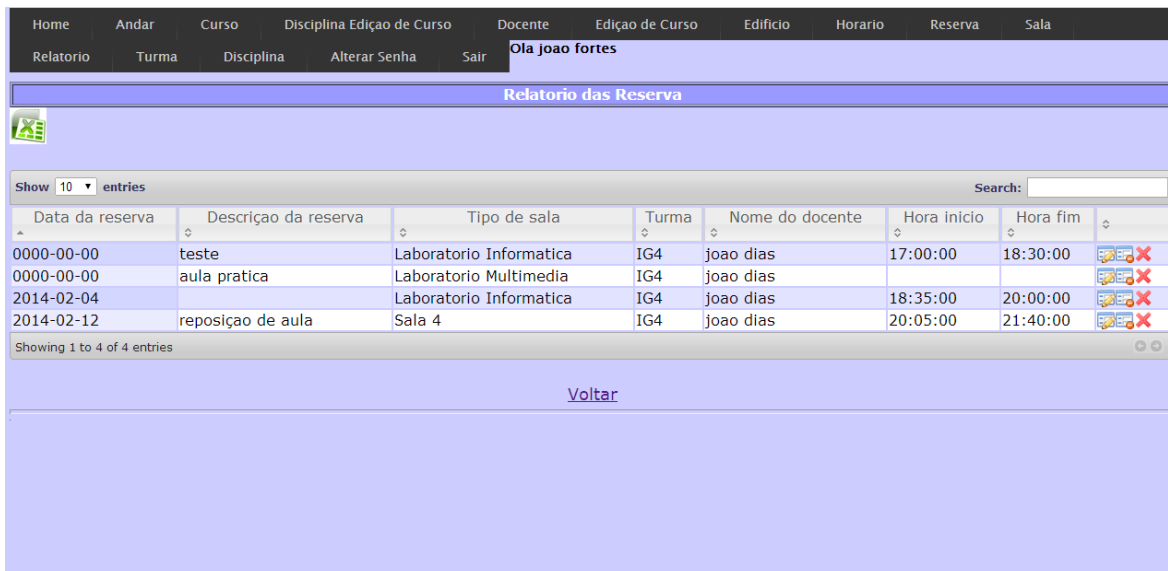














Figura 24- Interface gestão utilizadores

4.5.2.7 Relatório de gestão das reservas

A figura 25 mostra a utilização do relatório de gestão de reservas de salas. Neste relatório é possível utilizar os filtros, assim o relatório poderá ser separado por salas ou separado por períodos de data.



Data da reserva	Descrição da reserva	Tipo de sala	Turma	Nome do docente	Hora início	Hora fim	
0000-00-00	teste	Laboratorio Informatica	IG4	joao dias	17:00:00	18:30:00	  
0000-00-00	aula pratica	Laboratorio Multimedia	IG4	joao dias			  
2014-02-04		Laboratorio Informatica	IG4	joao dias	18:35:00	20:00:00	  
2014-02-12	reposição de aula	Sala 4	IG4	joao dias	20:05:00	21:40:00	  

Showing 1 to 4 of 4 entries

[Voltar](#)

Figura 25-Interface relatório reservas

Os relatórios podem ser impressos ou visualizados em tela. Todos os relatórios possuem filtros, o que o torna de fácil entendimento.

4.5.2.8 Interface principal utilizador gestor de salas

A figura n apresenta a tela principal de acesso “gestor de salas”. Nesta figura apresentam-se os menus de cadastros, reservas, relatório e alterar senha.



Figura 26-Interface principal gestor salas

Ao clicar no item de menu “Alterar senha”, apresenta-se uma interface em que o utilizador que estiver autenticado poderá proceder à alteração da sua senha (ver figura 27).

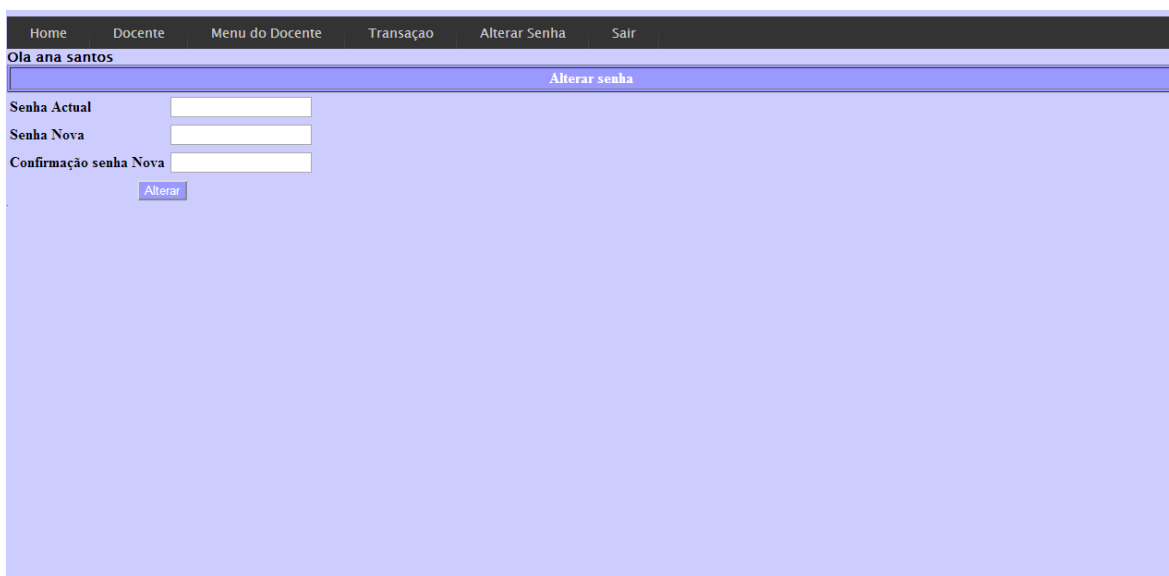


Figura 27-Interface alteração de senha

CAPITULO 5 - CONCLUSÃO

O desenvolvimento do sistema web para reservas de salas teve como intuito, fornecer à Universidade do Mindelo uma ferramenta computacional para efetuar a reserva de salas através da internet de forma simples e organizada. O agendamento de reservas realizado por meio deste sistema proporciona maior flexibilidade e eficiência no processo de reserva. A utilização desta ferramenta computacional elimina o uso do papel no processo de reservas e possibilita que o fluxo de informações seja mais dinâmico e eficaz.

As ferramentas utilizadas no desenvolvimento atenderam às necessidades do projeto. A linguagem PHP se apresentou como uma linguagem flexível, e o MySQL disponibilizou uma maneira consistente de armazenar e manipular os dados.

De modo geral, a realização deste trabalho proporcionou maior aprendizagem de desenvolvimento web e me permitiu aplicar os conhecimentos adquiridos no decorrer do curso para o desenvolvimento de um *software*.

Por ultimo, o ganho do aprendizado que este académico teve com a execução deste trabalho, pois houve a necessidade de adquirir mais conhecimentos do que possui até o início deste. Pois muitas vezes em desenvolvimento de uma aplicação o desenvolvedor se confronta com barreiras ou mais de uma saída, e isso faz com que o mesmo adquira mais conhecimentos e ao mesmo tempo entenda o que está vendo a para isso saber a melhor escolha a tomar.

Sugestões

Como sugestões para trabalhos futuros sugere-se os seguintes:

Disponibilizar um histórico de acessos para manter um acompanhamento da utilização do sistema pelos utilizadores.

Permitir a exportação de relatórios.

10 ANOS DE EMPENHO E DEDICAÇÃO

Disponibilizar um *chat*, permitindo a conversa instantânea entre docentes e administradores.

Sugere-se também a integração do sistema com um sistema para a formulação da grade horaria, de acordo com algumas restrições como a disponibilidade de docentes.

Bibliografia

ALECRIM, Emerson (2006). Conhecendo o apache.
<http://www.infowester.com/servapach.php> acesso em 28/11/2013 às 12:56.

CARRIÇO, António (1996). *Desenho de bases de dados*. Editora: Centro d tecnologias de informação.

CHIOSSI, Thelma e MORAES, Regina (2006). *Especificação de sistemas de software utilizando analise e projetos estruturados*. Editora Unicamp.

Criptografia MD5. Disponível em <http://www.devmedia.com.br/criptografia-md5/2944>
acesso em 28/02/2014 às 18:37

FERREIRA, Milton (2013). Segurança da informação e sistemas.
<http://www.apinfo.com/artigo81.htm> acesso em 07/12/2013 às 20:16

MARQUES, Joaquim e SERRÃO, Carlos (2007). *PHP5*. FCA Editora de Informática.

Méetodo Post. Disponível em
http://www.homehost.com.br/artigos/metodo_post_envio_de_dados_invisivel_em_php-035.html acesso em 16/03/2014 às 09:29

NUNES, Mauro e O'NEILL, Henrique (2004). *Fundamental de UML*. 4ª Edição. Editora FCA.

PACIEVITCH, yuri(2012). *HTML – Informatica – InfoEscola*. <http://www.infoescola.com/informatica/html/> acesso em 21/11/2013 às 15:19

PADUA, Paula (2009). *Engenharia de software: fundamentos, métodos e padrões*. 3ª Edição. Rio de Janeiro LTC.

PASCUTTI, Augusto (2013). Porque utilizar PHP. <http://www.slideshare.net> acesso em 24/11/2013 às 12:51

PISO, pedro (2012). *O que é e como usar o MySQL*. <http://www.techtudo.com.br/artigo/noticia/2012/04/o-que-e-e-como-usar-o-mysql.html> acesso em 21/11/2013 às 15:06.

PRADO, Juliana (2013). *Evolução da metodologia de desenvolvimento de sistemas*. <http://www.linhadecodigo.com.br/artigo/2108/evolucao-da-metodologia-do-desenvolvimento-de-sistemas.aspx> acesso em 29/12/2013 às 16:37

PRESSMAN, Roger (2002). *Engenharia de Software*. 5ª Edição. MC Graw Hill Editora.

REBELLO, Jair (2012). PHP Orientado a Objetos: MVC em PHP – Escola Criatividade. <http://www.escolacriatividade.com/php-orientado-a-objetos-mvc-em-php/> acesso em 08/12/2013 às 12:53.

SILVA, Alberto e VIDEIRA, Carlos (2005). *UML Metodologias e Ferramentas Case*. Editora Centro Atlântico.

STAIR, Ralph e REYNOLDS, George (2006). *Princípios de Sistemas de Informação*. 6ª Edição. Editora Thomson.

VALADARES, Diego (2013). *Pesquisa Operacional – Conceitos de pesquisa operacional*.
<http://www.ebah.com.br/content/ABAAABHioAL/pesquisa-operacional> acesso em
14/03/2014 às 11:18

ZENETI, Luiz (2003). *Sistemas de informação baseados na tecnologia web: Um estudo sobre o seu desenvolvimento*. <http://www.teses.usp.br/teses/disponiveis/12/12139/tde-14082003-104928/pt-br.php> acesso em 27/11/2013 às 12:08.

Apêndices

Apêndice A – Entrevista

ENTREVISTA

Entrevista submetida ao Coordenador do Curso de Informática de Gestão e Vice-Presidente da Universidade do Minelo para o levantamento de requisitos do Projeto de Final de Curso intitulado “Sistema de Controlo e Reserva de Salas da Universidade do Minelo”, cuja autora é Taís Silva Andrade.

1. Como é realizado o serviço de reserva de salas e quais documentos são utilizados?
2. Quantas salas possui a Universidade?
3. Como é feita a classificação de salas da Universidade?
4. Quantos computadores há em cada laboratório?
5. A respeito da reserva de salas haveria alguma restrição em relação a quantidade de salas e dias possíveis a serem reservados?
6. Quais serão as permissões padrão de acesso ao sistema?
7. Sobre o cadastro de utilizadores, quem poderá cadastrar quem? Os utilizadores poderão se auto-cadastrar?
8. O sistema deverá permitir que a reserva de salas seja feita por turno, dia ou por hora?
9. Como funciona o processo de reserva de salas atualmente?
10. O aluno poderá reservar salas?

Apêndice B – Descrição estruturada dos casos de uso

Neste apêndice são apresentados em detalhes os principais casos de uso do sistema previstos no diagrama na secção 4.3.1.

Na tabela 16 apresenta-se o caso de uso “Autenticar utilizador”

Caso de uso	Autenticar utilizador
Atores	Docente, Gestor salas e administrador
Objetivo	Identificar o tipo de utilizador
Pré-condições	Utilizador cadastrado
Pós-condições	Apresenta o menu do sistema de acordo com o tipo de utilizador identificado.
Cenário Principal	1-Utilizador informa seu e-mail e senha 2- Sistema valida os dados 3- Sistema exhibe diferentes opções do menu de acordo com o tipo de utilizador.

Tabela 16 - Descrição caso uso autenticar utilizador

Na tabela 17 apresenta-se o caso de uso “Cadastro Curso”.

Caso de uso	Cadastro curso
Atores	Gestor de salas
Objetivo	Cadastro dos cursos existentes na UM

10 ANOS DE EMPENHO E DEDICAÇÃO

Pré-condições	Gestor cadastrado
Pós-condições	Curso cadastrado
Cenário principal	<p>1-Gestor de salas abre o cadastro de curso</p> <p>2-Gestor de salas informa dados referentes ao curso</p> <p>3-Sistema armazena na base de dados e emite aviso de sucesso.</p>

Tabela 17 - Descrição caso uso cadastro curso

Na tabela 18 apresenta-se o caso de uso “Cadastro docente”

Caso de uso	Cadastro docente
Atores	Gestor de salas
Objetivo	Cadastro dos docentes que lecionam na UM
Pré-condições	Gestor de salas cadastrado
Pós-condições	Docente cadastrado e apto para autenticar-se no sistema
Cenário principal	<p>1-Gestor de salas abre o cadastro de docente</p> <p>2-Gestor de salas informa o nome do docente, e-mail e senha</p> <p>3- Sistema armazena na base de dados e emite aviso de sucesso.</p>

Tabela 18 - Descrição caso uso cadastro docente

Na tabela 19 apresenta-se o caso de uso “cadastro horário”

Caso de uso	Cadastro horário
Atores	Gestor de salas
Objetivo	Cadastro dos horários da UM
Pré-condições	Gestor de salas cadastrado
Pós-condições	Horário cadastrado
Cenário principal	1-Gestor de salas abre o cadastro de horário 2-Gestor de salas informa os dados do horário 3-Sistema armazena dados na base de dados e emite aviso de sucesso.

Tabela 19 - Descrição caso uso cadastro horário

Na tabela 20 apresenta-se o caso de uso cadastro disciplina

Caso de uso	Cadastro disciplina
Atores	Gestor de salas
Objetivo	Cadastro das disciplinas dos diferentes cursos
Pré-condições	Gestor de salas autenticado
Pós-condições	Disciplina cadastrada e vinculada ao curso

Cenário principal	<p>1-Gestor de salas abre o cadastro de disciplinas</p> <p>2-sistema lista os cursos cadastrados</p> <p>3-Gestor de salas escolhe um curso e informa o nome da disciplina</p> <p>4-Sistema armazena os dados e emite aviso de sucesso.</p>
--------------------------	--

Tabela 20 - Descrição caso uso cadastro disciplina

Na tabela 21 apresenta-se o caso de uso “cadastro sala”.

Caso de uso	Cadastro sala
Atores	Gestor de salas
Objetivo	Cadastro das diferentes salas disponíveis na UM
Pré-condições	Gestor de salas autenticado
Pós-condições	Sala cadastrada
Cenário principal	<p>1-Gestor de salas abre o cadastro de salas</p> <p>2-Gestor sala informa os dados da sala</p> <p>3-Sistema armazena na base de dados e emite mensagem de sucesso.</p>

Tabela 21 - Descrição caso uso cadastro sala

Na tabela 22 apresenta-se o caso de uso “emitir relatório por utilizador ou por período”.

Caso de uso	Emitir relatório por utilizador ou por período
Atores	Gestor de salas
Objetivo	Observar estatísticas de utilização
Pré-condições	Gestor de salas autenticado
Pós-condições	Relatório gerado
Cenário principal	<p>1-Gestor de salas solicita a opção de relatórios</p> <p>2-Gestor de salas seleciona entre utilizador ou período</p> <p>3-Sistema exibe lista de utilização por recurso.</p>

Tabela 22 - Descrição caso uso emitir relatório

A tabela 23 apresenta o caso de uso “efetuar reserva”.

Caso de uso	Efetuar reservas
Atores	Docente
Objetivo	Observar estatísticas de utilização
Pré-condições	Docente deve estar autenticado

10 ANOS DE EMPENHO E DEDICAÇÃO

Pós-condições	Reserva efetuada com sucesso
Cenário principal	1- Informa os dados da reserva 2- Sistema efetua a reserva 3- Sistema exibe mensagem de reserva efetuada.

Tabela 23 - Descrição caso uso efetuar reserva